



TUGAS AKHIR - TE 141599

## **PENJEJAKAN TARGET SENJATA OTOMATIS MENGUNAKAN METODE IMAGE BASED VISUAL SERVOING**

Zakki Muhammad  
NRP 2212100093

Dosen Pembimbing  
Dr. Ir. Djoko Purwanto, M. Eng

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





**FINAL PROJECT - TE 141599**

**TARGET TRACKING OF AUTOMATIC GUN USING  
IMAGE BASED VISUAL SERVOING**

Zakki Muhammad  
NRP 2212100093

Supervisor  
Dr. Ir. Djoko Purwanto, M. Eng

ELECTRICAL ENGINEERING DEPARTMENT  
Faculty of Industrial Technology  
Sepuluh Nopember Institute of Technology  
Surabaya 2017



## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan Penjejakan Target Senjata Otomatis Menggunakan Metode *Image Based Visual Servoing* adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan berlaku.

Surabaya, Januari 2017

Zakki Muhammad  
2212100093



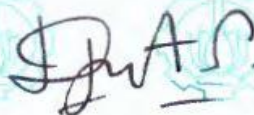
**PENJEJAKAN TARGET SENJATA OTOMATIS  
MENGUNAKAN METODE *IMAGE BASED*  
*VISUAL SERVOING***

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik  
Pada  
Bidang Studi Elektronika  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**

**Dosen Pembimbing**



**Dr. Ir. Djoko Purwanto, M.Eng.  
NIP. 196512111990021002**







# **PENJEJAKAN TARGET SENJATA OTOMATIS MENGUNAKAN METODE *IMAGE BASED VISUAL SERVOING***

**Nama : Zakki Muhammad**  
**Pembimbing : Dr. Ir. Djoko Purwanto, M. Eng.**

## **ABSTRAK**

Senjata yang mampu menjejak dan menembak sasaran secara otomatis merupakan senjata yang telah diberi sistem penjejakan otomatis dengan menggunakan teknologi *image processing*. Sistem tersebut bekerja dengan melakukan penjejakan terhadap target yang telah dipilih untuk menjadi sasaran tembak oleh operator. Sistem ini digunakan untuk mengurangi korban manusia di daerah yang mengalami banyak ancaman dan teror

Sistem ini tidak akan bekerja untuk menjejak target apabila tidak ada kamera. Oleh karena itu kamera diletakkan pada senjata guna untuk mendapatkan info mengenai sasaran. Sasaran dipilih untuk selanjutnya dilacak. Data dari pengambilan gambar tersebut diolah untuk memerintahkan senjata agar segera menjejak sasaran yang telah dipilih. Operator bertugas sebagai pemilih sasaran yang perlu dijejak atau tidak perlu.

Hasil dari pengujian yang telah dilakukan pada tugas akhir ini adalah sistem mampu menjejak target manusia dengan sasaran bergerak maupun diam dengan rentang waktu 2 detik sampai 5 detik. Kesalahan penjejakan target untuk gerak horizontal mencapai 23% dengan rata-rata kesalahan gerak tengah ke kiri sebesar 6,716%, kiri ke kanan sebesar 9,027%, tengah ke kanan sebesar 17,805%, dan kanan ke kiri sebesar 9,478%, sedangkan untuk gerak vertikal mencapai 16% dengan rata-rata kesalahan gerak tengah ke atas sebesar 8,559%, atas ke bawah sebesar 10,788%, tengah ke bawah sebesar 14,955%, dan bawah ke atas sebesar 11,247% Penjejakan target dengan metode *optical flow* bekerja buruk pada kondisi sasaran bergerak terlalu cepat.

Kata kunci: pengolah citra, kamera, senjata otomatis

*#Halaman ini sengaja dikosongkan#*

# ***TARGET TRACKING OF AUTOMATIC GUN USING IMAGE BASED VISUAL SERVOING***

**Name : Zakki Muhammad**  
**Supervisor : Dr. Ir. Djoko Purwanto, M. Eng.**

## ***ABSTRACT***

Weapon that can track and target shooting automatic weapons are granted automatic tracking system using image processing technology. The system works by tracking the target that has been selecte to be a target by the operator. This system is used to reduced the human toll in area that experience a lot of threats and error.

This system will not work to track the target when there is no camera. Therefore, the camera is placed on the weapon used to get info on target. Selected target to the next track. Data from image capture are processed to order weapons to immediately track the targets. Operator on duty as a voter targets to be tracked or not to be tracked.

The results of the testing that has been done in this final project is a system able to track human targets with moving or stationary target with a span of 2 seconds to 5 seconds. Error tracking the target's horizontal motion reached 23% with an average error motion of the center to the left at 6.716%, left to right amounting to 9.027%, the center to the right at 17.805%, and the right to the left at 9.478%, while for vertical motion reaches 16 % with an average error of motion center to top at 8.559%, top to bottom at 10.788%, center to bottom at 14.955%, and bottom to top at 11.247%. Target with a trace of an optical flow method works poorly on condition of target moves too fast.

**Keywords:** image processing, camera, automatic weapons

*# Halaman ini sengaja dikosongkan #*

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan rahmat-Nya penulis dapat menyelesaikan Tugas Akhir ini dengan judul :

**Penjejakan Target Senjata Otomatis Menggunakan Metode *Image Based Visual Servoing***

Tugas Akhir ini merupakan persyaratan dalam menyelesaikan pendidikan program Strata-Satu di Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dibuat berdasarkan teori-teori yang didapat selama mengikuti perkuliahan, berbagai literatur penunjang dan pengarahan dosen pembimbing dari awal hingga akhir pengerjaan Tugas Akhir ini.

Pada kesempatan ini, penulis ingin berterima kasih kepada pihak-pihak yang membantu pembuatan tugas akhir ini, khususnya kepada:

1. Bapak, Ibu, adik serta seluruh keluarga yang memberikan dukungan baik moril maupun materiil.
2. Dr. Ir. Djoko Purwanto, M. Eng. selaku dosen pembimbing atas bimbingan dan arahan selama penulis mengerjakan tugas akhir ini.
4. Tasripan, IR. MT selaku Koordinator Bidang Studi Elektronika.
5. Dr. Eng. Ardyono Priyadi, ST., M. Eng selaku Ketua Jurusan Teknik Elektro ITS Surabaya.
6. Seluruh dosen bidang studi elektronika.
7. Teman-teman laboratorium Elektronika yang tidak dapat disebutkan satu-persatu, telah membantu proses pengerjaan tugas akhir ini.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang perlu diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut.

Akhirnya penulis berharap agar Tugas Akhir ini dapat memberikan manfaat yang sebesar-besarnya bagi semua pihak. Penulis juga berharap supaya Tugas Akhir ini dapat menjadi aplikasi yang lebih bermanfaat.

*# Halaman ini sengaja dikosongkan #*

## DAFTAR ISI

ABSTRAK .....	i
ABSTRACT .....	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL .....	xi
BAB I PENDAHULUAN .....	1
1.1    Latar Belakang.....	1
1.2    Perumusan Masalah.....	2
1.3    Batasan Masalah.....	2
1.4    Tujuan.....	2
1.5    Metodologi .....	2
1.6    Sistematika Penulisan .....	4
1.7    Relevansi .....	4
BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG ...	5
2.1    Citra (Image) Digital .....	5
2.1.1    Citra RGB.....	6
2.1.2    Citra Biner .....	8
2.1.3    Citra Grayscale .....	9
2.2    Pengolahan Citra (Image Processing).....	11
2.2.1    Metode Optical flow .....	11
2.2.2    OpenCV .....	13
2.2.3    Qt creator .....	13
2.3    Senjata .....	14
2.3.1    Machine Gun .....	14
2.3.2    Submachine Gun .....	15
2.3.3    Assault Rifle .....	15

2.4	Pulse Width Modulation (PWM).....	16
2.5	Driver Motor.....	17
<b>BAB III PERANCANGAN SISTEM.....</b>		<b>19</b>
3.1	Perancangan Software .....	20
3.1.1	Pengambilan Citra .....	20
3.1.2	Grayscale .....	21
3.1.3	Optical flow Lucas-Kanade .....	22
3.1.4	Fuzzy Logic Controller.....	23
3.2	Perancangan Hardware .....	27
3.2.1	Arduino Nano .....	27
3.2.2	Driver motor .....	29
3.2.3	Motor DC.....	30
3.2.4	Senjata.....	31
3.2.5	Kamera.....	32
3.2.6	Notebook.....	32
<b>BAB IV PENGUJIAN DAN ANALISIS .....</b>		<b>33</b>
4.1	Sistem Penjejak.....	33
4.2	Pengujian Motor .....	34
4.3	Pengujian Serial Port USB to TTL .....	35
4.4	Pengujian Optial Flow Pada Siang Hari .....	36
4.5	Pengujian Optical Flow Pada Malam Hari .....	38
4.6	Pengujian Performa Sistem.....	39
<b>BAB V KESIMPULAN DAN SARAN .....</b>		<b>49</b>
5.1	Kesimpulan.....	49
5.2	Saran .....	49
<b>DAFTAR PUSTAKA.....</b>		<b>51</b>
<b>LAMPIRAN .....</b>		<b>53</b>
<b>BIODATA PENULIS .....</b>		<b>73</b>



## DAFTAR GAMBAR

Gambar 2.1 RGB pada citra digital .....	7
Gambar 2.2 RGB pada citra digital .....	7
Gambar 2.3 Konsep piksel RGB .....	8
Gambar 2.4 Gradasi citra biner .....	8
Gambar 2.5 Citra biner .....	9
Gambar 2.6 Konsep piksel citra biner .....	9
Gambar 2.7 Citra grayscale .....	10
Gambar 2.8 Konsep piksel citra grayscale .....	10
Gambar 2.9 Pergerakan titik.....	11
Gambar 2.10 Piramida Optical flow .....	12
Gambar 2.11 Logo OpenCV.....	13
Gambar 2.12 Logo Qt dan slogannya.....	14
Gambar 2.13 Machine gun .....	14
Gambar 2.14 Submachine gun .....	15
Gambar 2.15 Assault Rifle .....	15
Gambar 2.16 Hasil PWM berbagai duty cycle .....	16
Gambar 2.17 Rangkaian driver motor dc .....	17
Gambar 3.1 Sistem dengan pemilihan target.....	19
Gambar 3.2 Diagram blok sistem .....	20
Gambar 3.3 Pengambilan citra 640x480 .....	20
Gambar 3.4 Hasil citra grayscale pada video capture.....	21
Gambar 3.5 Penandaan target yang dipilih operator .....	23
Gambar 3.6 Diagram blok penjejak dengan fuzzy .....	24
Gambar 3.7 Board arduino nano .....	28
Gambar 3.8 Driver motor BTN7970 .....	29
Gambar 3.9 Motor horizontal .....	31
Gambar 3.10 Motor vertikal.....	31

Gambar 3.11 SIG552 .....	31
Gambar 3.12 Kamera Logitech C920 .....	32
Gambar 4.1 Sistem penjejak .....	33
Gambar 4.2 Sistem penjejak .....	34
Gambar 4.3 USB to TTL .....	35
Gambar 4.4 Pengecekan port USB to TTL.....	36
Gambar 4.5 Uji optical flow .....	36
Gambar 4.6 Uji Optical flow .....	37
Gambar 4.7 Uji optical flow .....	37
Gambar 4.8 Uji coba optical flow malam hari.....	38
Gambar 4.9 Uji coba optical flow malam hari.....	39

## DAFTAR TABEL

Tabel 3.1 Fuzzy Rule Base Horizontal .....	25
Tabel 3.2 Fuzzy Rule Base Vertikal .....	26
Tabel 3.3 Penggunaan port arduino nano .....	29
Tabel 3.4 Tabel kebenaran driver motor Vertikal .....	30
Tabel 3.5 Tabel kebenaran driver motor Horizontal .....	30
Tabel 4.1 Pengujian motor DC horizontal.....	34
Tabel 4.2 Pengujian motor DC vertikal.....	35
Tabel 4.3 Ilustrasi pergerakan target untuk uji coba .....	40
Tabel 4.4 Pengujian performa sistem gerak horizontal.....	42
Tabel 4.5 Ilustrasi pergerakan bola.....	40
Tabel 4.6 Pengujian performa sistem gerak vertikal.....	47

*# Halaman ini sengaja dikosongkan #*

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam bidang militer, penggunaan tenaga manusia untuk melakukan tugas yang membutuhkan tingkat akurasi dan presisi yang tinggi perlahan dikurangi dan digantikan dengan suatu sistem yang telah diotomasi. Contoh dari hal yang membutuhkan akurasi dan presisi yang tinggi adalah penjejakkan suatu target tembak.

Senjata-senjata masa kini telah banyak diciptakan dengan kemampuan penembakan yang tinggi. Senjata yang digunakan semakin canggih, akan tetapi sulit mendapatkan tingkat akurasi dan presisi yang tinggi diakibatkan adanya faktor manusia. Keterbatasan kemampuan manusia untuk menggunakan senjata sudah tentu mengurangi tingkat akurasi dan presisi penembakan target. Manusia dapat kelelahan di suatu waktu, oleh karena itu dengan senjata canggih sekali pun akan sulit untuk mendapatkan hasil penembakan yang akurat.

Perkembangan teknologi otomatis tidak dapat dibendung tidak terkecuali dalam bidang militer. Hampir semua serba otomatis, termasuk senjata. Senjata saat ini sudah banyak diotomasi untuk mendapatkan tingkat penembakan yang akurat dan presisi. Senjata yang telah diotomasi akan mencegah terjadinya kesalahan penembakan atau ketidakakuratan penembakan.

Senjata digunakan untuk menangani perkara besar, perang merupakan golongan perkara besar yang memerlukan senjata. Konflik perang sedang terjadi di mana-mana, baik di negara Eropa, Asia, maupun Afrika. Perang paling brutal terjadi sepanjang 2014. Korban terus berjatuhan dan menewaskan puluhan ribu orang tak bersalah. Seperti diberitakan IB Times, Sabtu (21/3/2015), sejumlah sumber, termasuk militer Amerika Serikat (AS), Perserikatan Bangsa-Bangsa (PBB), dan Observatorium untuk Hak Asasi Manusia, terus menghitung negara yang korbannya paling banyak. Konflik di India terjadi sejak 2013. Pada 2014, sumber menjelaskan sebanyak 976 orang tewas karena perang tersebut. Seperti India, konflik di Republik Demokratik Kongo terjadi pada 2013. Sebanyak 1.235 orang tewas akibat perang tersebut. Namun, jumlah itu jauh lebih sedikit dari sebelumnya yang mencapai 1.976 orang tewas dan masih banyak konflik perang yang terjadi diberbagai negara.

Oleh karena itu, dalam tugas akhir ini dirancang sebuah sistem pelacakan dan penembakan yang dapat diterapkan pada daerah pertahanan guna mengatasi teror dan ancaman terhadap warga sipil yang tidak bersalah. Dengan pemanfaatan teknologi *image processing* yang diolah pada *processing unit* sistem ini mampu melakukan penjejakan dan penguncian sasaran secara otomatis. Diharapkan sistem ini mampu menstabilkan keakuratan penembakan dan mencegah timbulnya korban jiwa pada daerah yang memiliki konflik.

## **1.2 Perumusan Masalah**

Perumusan masalah pada tugas akhir ini meliputi:

1. Bagaimana cara untuk mendapatkan target?
2. Bagaimana cara untuk mengenali target?
3. Bagaimana cara untuk melakukan penjejakan target?

## **1.3 Batasan Masalah**

Batasan masalah pada tugas akhir ini meliputi:

1. Lokasi pengambilan citra memiliki cahaya yang cukup
2. Sasaran yang dijejak dipilih secara manual

## **1.4 Tujuan**

Tujuan dari tugas akhir sebagai berikut:

1. Sistem mampu mengenali target dengan cepat
2. Sistem mampu menjejak gerakan target
3. Sistem dapat digunakan operator dengan mudah

## **1.5 Metodologi**

Langkah-langkah yang dilakukan pada tugas akhir ini sebagai berikut:

### **1. Studi Literatur**

Pada tahap ini dilakukan pencarian teori-teori dasar yang mendukung untuk penyelesaian tugas akhir. Teori-teori dasar dapat didapatkan dari jurnal ilmiah, buku, dan sumber internet terpercaya.

## **2. Perancangan *software***

Pada tahap ini dirancang perangkat lunak (*Software*) yang berisi kode agar sistem ini mampu mengenali target dan mampu menjejaknya. Untuk pelacakan dan penguncian sasaran, dapat dilakukan dengan membandingkan error dari jarak target sekarang menuju titik tengah (titik tembak) dan error pada frame dari image yang diterima saat itu dengan error pada frame sebelumnya yang ditangkap oleh kamera yang digunakan sebagai referensi sasaran

## **3. Perancangan *hardware***

Pada tahap ini dirancang mekanik dari sistem yang mampu menjalankan tugas pergerakan untuk menjejak target. Hardware terdiri dari kamera, motor dc, senjata danudukan senjata. Kamera digunakan untuk pengambilan citra, motor dc digunakan untuk menggerakkan senjata agar mampu menjejak. Terdapat dua motor dc yang digunakan, satu untuk gerak horizontal dan satunya lagi untuk gerak vertikal.

## **4. Pengujian sistem**

Pada tahap ini dilakukan pengujian kerja dari sistem yang telah dibuat. Pengujian ini dilakukan untuk mendapatkan informasi bahwa hasil *software* yang telah dibuat dapat terhubung dengan *hardware*.

Pengujian dilakukan dengan mendeteksi keberadaan manusia setelah itu mengunci sasaran setelah itu barulah sistem melakukan penjejakan terhadap sasaran yang telah dipilih.

## **5. Analisa**

Pada tahap ini sistem yang telah dibuat akan dianalisa untuk mendapatkan informasi mengenai hasil yang didapatkan agar sesuai dengan tujuan untuk menjejak target secara tepat

## **6. Penulisan laporan akhir**

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka dan teori penunjang, perancangan sistem, pengujian, dan penutup.

## 1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

- **BAB I : PENDAHULUAN**  
Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.
- **BAB II : TINJAUAN PUSTAKA DAN TEORI PENUNJANG**  
Bab ini menjelaskan tentang teori penunjang dan pustaka yang dibutuhkan dalam pengerjaan tugas akhir ini. Dasar teori yang menunjang pada tugas akhir ini meliputi pengenalan citra digital, pengolahan citra (*Image processing*) yang terdiri dari *haar cascade* dan teori *optical flow*, *pulse width modulation (PWM)*, *driver motor*.
- **BAB III : PERANCANGAN SISTEM**  
Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk sistem penjejak target senjata otomatis
- **BAB IV : PENGUJIAN**  
Pada bab ini dilakukan pengujian kerja dari sistem yang telah dibuat untuk memastikan kesesuaian hubungan antara *software* dan *hardware*
- **BAB V : PENUTUP**  
Pada bagian ini terdapat kesimpulan dari hasil percobaan dan analisis kerja dari sistem sekaligus mengandung saran-saran untuk pengembangan sistem menjadi lebih baik.

## 1.7 Relevansi

Hasil yang diperoleh dari tugas akhir ini diharapkan mampu mengurangi jumlah korban akibat peperangan serta menjaga kestabilan tingkat akurasi dari penembakan sasaran pada daerah pertahanan. Untuk lebih jauh lagi, sistem ini mampu bekerja tanpa bantuan operator sehingga hanya sebuah sistem dari teknologi mutakhir sudah mampu menangani masalah daerah konflik untuk pertahanan.



## **BAB II**

### **TINJAUAN PUSTAKA DAN TEORI PENUNJANG**

Teori penunjang dalam bab ini menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem yang akan dibuat pada tugas akhir ini. Sedangkan tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis-penulis sebelumnya.

#### **2.1 Citra (Image) Digital**

Citra atau Image merupakan istilah lain dari gambar, yang merupakan informasi berbentuk visual[1][2]. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Ketika sumber cahaya menerangi objek, objek memantulkan kembali sebagian cahaya tersebut. Pantulan ini ditangkap oleh alat-alat pengindera optik, misalnya mata manusia, kamera, scanner dan sebagainya. Bayangan objek tersebut akan terekam sesuai intensitas pantulan cahaya. Ketika alat optik yang merekam pantulan cahaya itu merupakan mesin digital, misalnya kamera digital, maka citra yang dihasilkan merupakan citra digital. Pada citra digital, kontinuitas intensitas cahaya dikuantisasi sesuai resolusi alat perekam.

Suatu citra adalah fungsi intensitas 2 dimensi  $f(x, y)$ , dimana  $x$  dan  $y$  adalah koordinat spasial dan  $f$  pada titik  $(x, y)$  merupakan tingkat kecerahan (brightness) suatu citra pada suatu titik. Citra digital adalah citra  $f(x, y)$  yang telah dilakukan digitalisasi baik koordinat area maupun brightness level. Nilai  $f$  di koordinat  $(x, y)$  menunjukkan brightness atau grayness level dari citra pada titik tersebut.

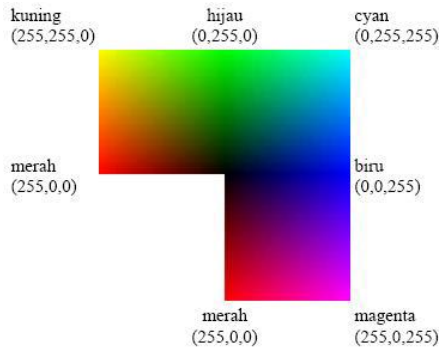
Citra Digital adalah representasi dari sebuah citra dua dimensi sebagai sebuah kumpulan nilai digital yang disebut elemen gambar atau piksel. Piksel adalah elemen terkecil yang menyusun citra dan mengandung nilai yang mewakili kecerahan dari sebuah warna pada sebuah titik tertentu. Umumnya citra digital berbentuk persegi panjang atau bujur sangkar (pada beberapa sistem pencitraan ada pula yang berbentuk segienam) yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dalam banyaknya piksel sehingga ukuran citra selalu bernilai bulat. Setiap piksel memiliki koordinat sesuai posisinya dalam

citra. Koordinat ini biasanya dinyatakan dalam bilangan bulat positif, yang dapat dimulai dari 0 atau 1 tergantung pada sistem yang digunakan. Setiap piksel juga memiliki nilai berupa angka digital yang merepresentasikan informasi yang diwakili oleh piksel tersebut. Format data citra digital berhubungan erat dengan warna. Pada kebanyakan kasus, terutama untuk keperluan penampilan secara visual, nilai data digital merepresentasikan warna dari citra yang diolah. Format citra digital yang banyak dipakai adalah Citra Biner (monokrom), Citra Skala Keabuan (gray scale), Citra Warna (true color), dan Citra Warna Berindeks.

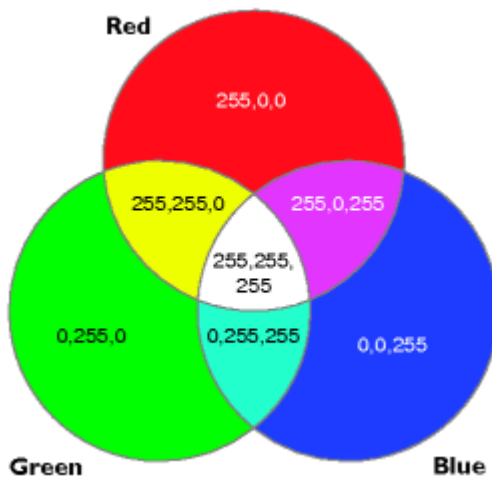
### **2.1.1 Citra RGB**

Suatu citra dalam model RGB terdiri dari tiga bidang citra yang saling lepas, masing-masing terdiri dari warna utama: merah, hijau dan biru. Suatu warna dispesifikasikan sebagai campuran sejumlah komponen warna utama. Gambar dibawah menunjukkan bentuk geometri dari model warna RGB untuk menspesifikasikan warna menggunakan sistem koordinat kartesian. RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak  $256 \times 256 \times 256 = 1677726$  jenis warna. Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang dimensi 3 yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen-x, komponen-y dan komponen-z.

Bentuk Representasi warna dari sebuah citra digital dapat dilihat pada gambar:

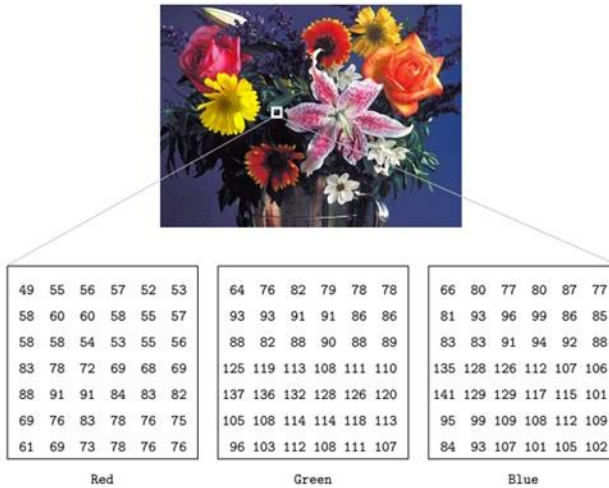


**Gambar 2.1** RGB pada citra digital



**Gambar 2.2** RGB pada citra digital

Misalkan sebuah vektor dituliskan sebagai  $r=(x,y,z)$ . Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(ed), G(reen), B(lue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB(30, 75, 255). Putih = RGB(255,255,255), sedangkan untuk hitam= RGB(0,0,0).[6]. Berikut merupakan konsep piksel pada citra RGB:



**Gambar 2.3** Konsep piksel RGB

### 2.1.2 Citra Biner

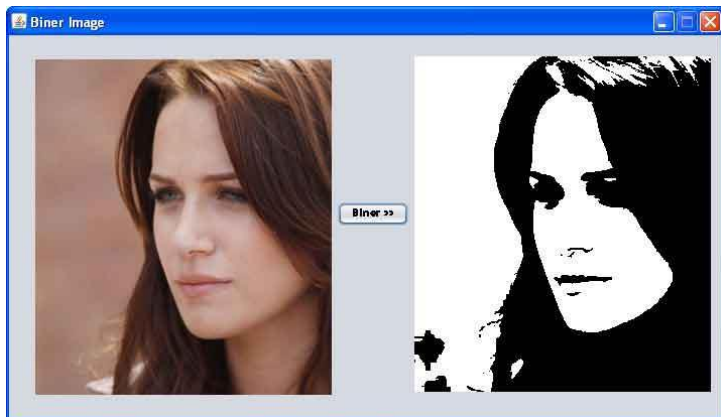
Citra biner adalah citra dimana piksel-piksennya hanya memiliki dua buah nilai intensitas yaitu bernilai 0 dan 1 dimana 0 menyatakan warna latar belakang (background) dan 1 menyatakan warna tinta/objek (foreground) atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih. Citra biner diperoleh dari nilai citra threshold sebelumnya.



**Gambar 2.4** Gradasi citra biner

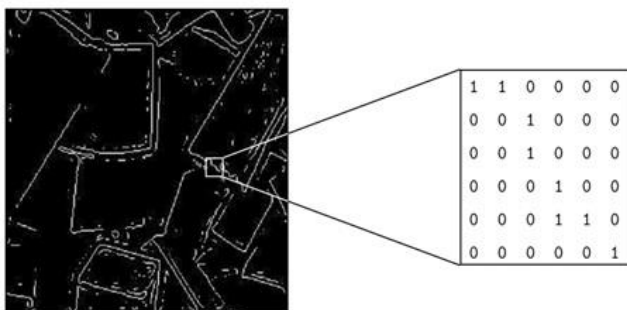
Meskipun saat ini citra berwarna lebih disukai karena memberi kesan yang lebih kaya daripada citra biner, namun tidak membuat citra biner mati. Pada beberapa aplikasi citra biner masih tetap dibutuhkan, misalnya citra logo instansi (yang hanya terdiri atas warna hitam dan putih), citra kode batang (bar code) yang tertera pada label barang, citra hasil pemindaian dokumen teks, dan sebagainya. Proses awal yang dilakukan dalam menganalisis objek di dalam citra biner adalah

segmentasi objek. Proses segmentasi bertujuan mengelompokkan pixel-pixel objek menjadi wilayah (region) yang merepresentasikan objek.



**Gambar 2.5** Citra biner

Berikut merupakan konsep piksel pada citra biner:



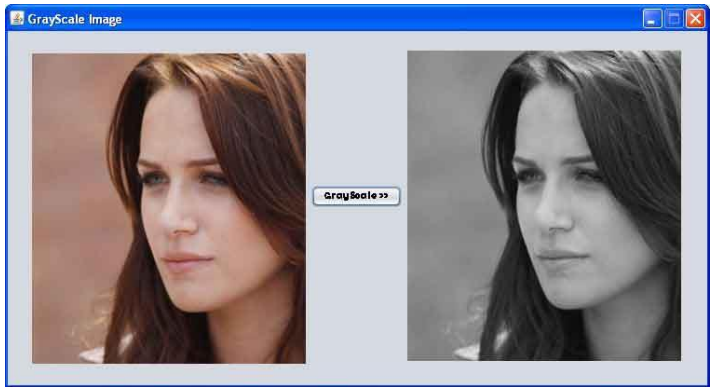
**Gambar 2.6** Konsep piksel citra biner

### 2.1.3 Citra *Grayscale*

Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra grayscale berbeda dengan citra "hitam-putih", dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu "hitam" dan "putih" saja. Pada citra

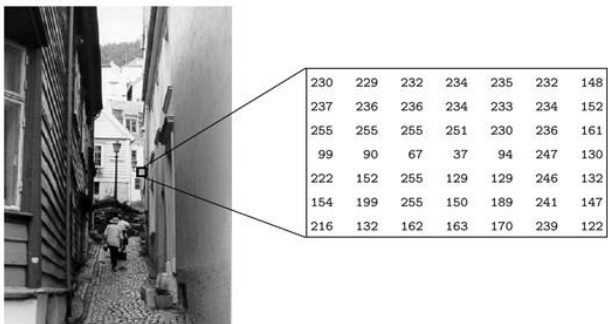
grayscale warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra grayscale seringkali merupakan perhitungan dari intensitas cahaya pada setiap pixel pada spektrum elektromagnetik single band.

Citra grayscale disimpan dalam format 8 bit untuk setiap sample pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra grayscale dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B.



**Gambar 2.7** Citra *grayscale*

Berikut merupakan konsep piksel pada citra *grayscale*:

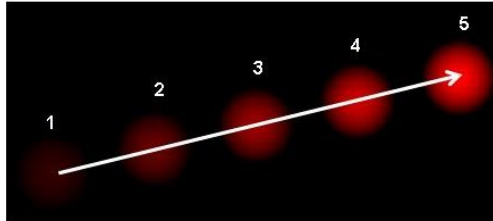


**Gambar 2.8** Konsep piksel citra *grayscale*

## 2.2 Pengolahan Citra (Image Processing)

### 2.2.1 Metode *Optical flow*

*Optical flow* merupakan pola dari gerak suatu objek gambar di antara dua frame yang disebabkan oleh pergerakan objek atau kamera. Ini dalam vektor 2D di mana setiap vektor merupakan perpindahan vektor yang menunjukkan pergerakan titik dari frame pertama ke frame kedua.



**Gambar 2.9** Pergerakan titik

Gambar tersebut menunjukkan 5 frame konsekutif. Panah menunjukkan perubahan vektornya. *Optical flow* memiliki banyak area fokus seperti:

- *Structure from Motion*
- *Video Compression*
- *Video Stabilization*

*Optical flow* bekerja pada asumsi:

- Intensitas piksel dari sebuah objek yang tidak berubah di antara beberapa konsekutif frame
- Piksel disekitar memiliki pergerakan yang sama

Dalam *Optical flow* digunakan metode Lucas-Kanade

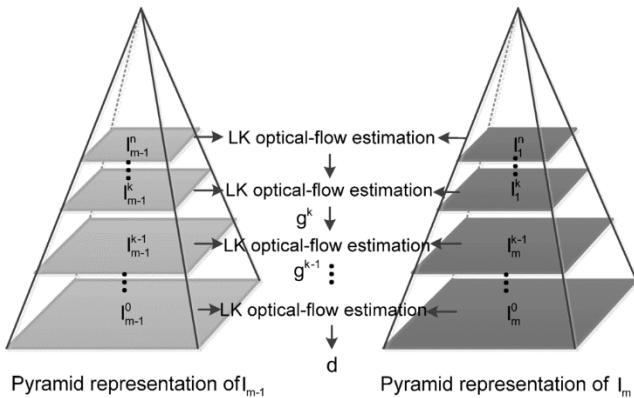
#### 2.2.1.1 Lucas-Kanade

Algoritma Lucas-Kanade, pertama kali diajukan pada tahun 1981. Pada awalnya algoritma ini adalah sebuah usaha untuk mencari teknik registrasi citra yang cepat dengan memanfaatkan gradient intensitas spasial. Pada perkembangannya, algoritma ini kemudian menjadi salah satu algoritma *optical flow* yang penting. Berbeda dengan algoritma Horn-Schunk yang bekerja berbasis pada keseluruhan citra, algoritma ini bekerja berdasar pada informasi lokal yang diturunkan dari window kecil (*patch*) disekeliling titik yang diperhitungkan dalam proses. Kelemahan digunakan window lokal kecil pada algoritma Lucas-Kanade

adalah tidak terdeteksinya gerakan-gerakan yang besar karena gerakan-gerakan tersebut jatuh diluar window. Akibatnya terjadi miss pada gerakan yang besar. Permasalahan ini kemudian dapat diatasi dengan mengimplementasikan penyelesaian dengan prinsip piramida, yaitu pyramidal Lucas-Kanade. Prinsip ini merupakan penyelesaian berdasar iterasi dari level detil citra paling rendah hingga level detil citra paling tinggi.

### 2.2.1.2 Piramida Lucas-Kanade

Penyelesaian algoritma *Lucas-Kanade* dengan pendekatan piramida, atau yang umum disebut dengan *Pyramidal Lukas-Kanade* diajukan pertama kali oleh Jean-Yves (Bouguet, pada tahun 2000)[4]. Pendekatan ini menggunakan prinsip piramida, yaitu bekerja mulai dari detil citra paling rendah hingga detil citra paling tinggi. Tujuannya adalah agar gerakan-gerakan yang “besar” dapat ikut diperhitungkan. Sementara asumsi yang digunakan pada algoritma *Lucas-Kanade* adalah gerakan yang “kecil” dan koheren, sehingga tidak dapat menangkap gerakan yang “besar”. Solusi untuk dapat menangkap gerakan yang “besar” pada algoritma *Lucas-Kanade* adalah dengan menggunakan *window* yang besar. Tetapi, akibat dari penggunaan *window* yang besar sering kali membuat gerakan yang ditangkap adalah gerakan yang tidak koheren. Algoritma *Pyramidal Lukas-Kanade* dapat menyelesaikan permasalahan tersebut tanpa menghilangkan asumsi gerakan-gerakan yang koheren.

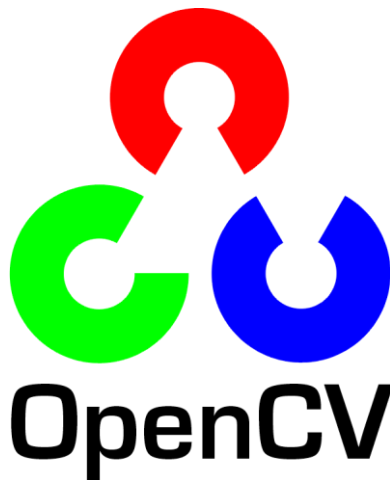


**Gambar 2.10** Piramida *Optical flow*



### 2.2.2 OpenCV

OpenCV (Open source Computer Vision) merupakan sebuah library dari fungsi programming untuk real-time computer vision. OpenCV dapat dimanfaatkan oleh program-program lainnya (seperti C++, C dan Python) untuk melakukan pengambilan, pengolahan serta penampilan data gambar, baik dalam bentuk image dan video maupun real-time video. OpenCV bersifat Open Source (dapat digunakan secara bebas) baik untuk akademik maupun untuk komersil, yang dapat berjalan dengan operating system Windows, Linus, Android dan Mac.



**Gambar 2.11** Logo OpenCV

### 2.2.3 *Qt creator*

Qt merupakan framework C++ yang komperhensif untuk mengembangkan aplikasi GUI (Graphical User Interface) lintas-platform menggunakan pendekatan “tuliskan sekali, compile di mana saja”[17]. Dengan Qt, programmer yang sudah membuat program user interface di platform Windows bisa meng-compile program tersebut di Linux.



**Gambar 2.12** Logo Qt dan slogannya

## **2.3 Senjata**

Senjata adalah suatu alat yang digunakan untuk melukai, membunuh, atau menghancurkan suatu benda. Senjata dapat digunakan untuk menyerang maupun untuk mempertahankan diri, dan juga untuk mengancam dan melindungi. Apapun yang dapat digunakan untuk merusak (bahkan psikologi dan tubuh manusia) dapat dikatakan senjata. Senjata bisa sederhana seperti pentungan atau kompleks seperti peluru kendali balistik.

### **2.3.1 Machine Gun**

*Machine Gun* (senapan mesin) adalah senjata api otomatis dipasang pada titik atau dibawa secara *portable* oleh setiap orang. Dirancang untuk menembak peluru secara berurutan dari sabuk amunisi atau magasin dan memiliki tingkat kecepatan menembak pada 3-1800 putaran per menit dan memiliki jarak efektif yang jauh bergantung dari kaliber yang digunakan. FN Herstal MINIMI MK3.



**Gambar 2.13** *Machine gun*

### 2.3.2 *Submachine Gun*

Submachine Gun adalah sebuah senjata api yang menggabungkan kemampuan menembak otomatis senapan mesin dengan amunisi pistol. Konsep senjata api seperti ini pertama kali dicoba pada tahun 1900-an, yaitu pistol yang diberi popor dan menembak secara otomatis



**Gambar 2.14** *Submachine gun*

### 2.3.3 *Assault Rifle*

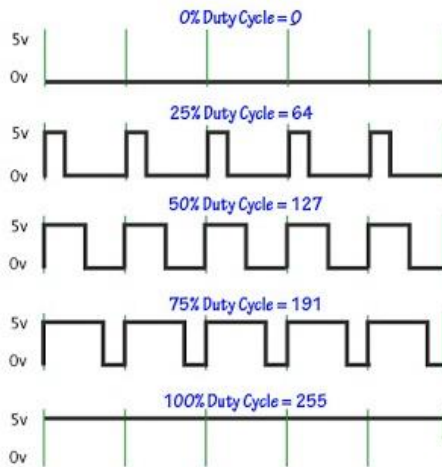
*Assault Rifle* (Senapan serbu) adalah senjata api otomatis yang merupakan senapan laras panjang atau karabin, yang memiliki pilihan tembakan (*selective-fire*), dan menggunakan amunisi kaliber menengah. Senapan serbu masuk dalam kategori di antara senapan mesin ringan, yang berfungsi untuk menembak secara full-otomatis sebagai senjata pendukung, dan *submachine gun*, senjata otomatis yang menggunakan peluru ukuran peluru pistol sebagai senjata api jarak dekat. Senapan serbu sudah menjadi persenjataan standar untuk tentara modern, menggantikan senapan laras panjang era Perang Dunia II.



**Gambar 2.15** *Assault Rifle*

## 2.4 Pulse Width Modulation (PWM)

PWM ( *Pulse Width Modulation*) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (*Duty cylce*) dengan nilai amplitudo dan frekuensi yang tetap. Satu siklus pulsa merupakan kondisi *driver* kemudian berada di zona transisi ke kondisi *low*. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi.



**Gambar 2.16** Hasil PWM berbagai *duty cycle*

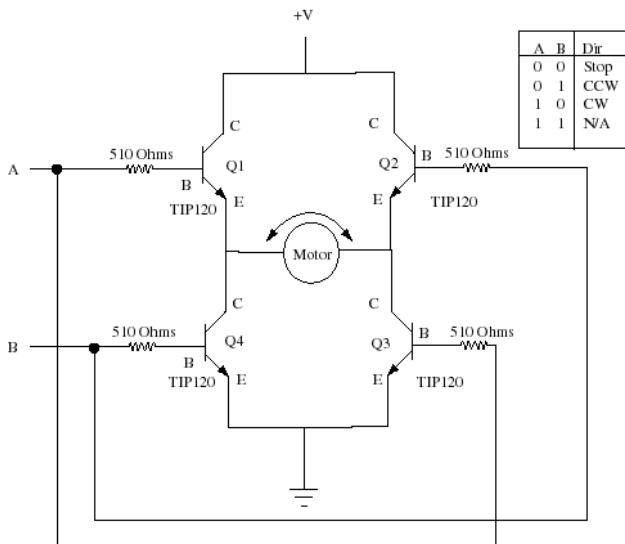
*Duty cylce* merupakan representasi dari kondisi logika *driver* dalam suatu periode sinyal dan dinyatakan dalam bentuk (%) dengan range 0% sampai 100%, sebagai contoh jika sinyal berada dalam kondisi *driver* terus menerus artinya memiliki *duty cycle* sebesar 100%. Jika waktu sinyal keadaan *driver* sama dengan keadaan *low* maka sinyal mempunyai *duty cycle* sebesar 50%.

Aplikasi penggunaan PWM biasanya ditemui untuk pengaturan kecepatan motor dc, pengaturan cerah/redup LED, dan pengendalian sudut pada motor servo. Contoh penggunaan PWM pada pengaturan kecepatan motor dc semakin besar nilai *duty cycle* yang diberikan maka akan berpengaruh terhadap cepatnya putaran motor. Apabila nilai *duty cylce*-nya kecil maka motor akan bergerak lambat.

## 2.5 Driver Motor

*Driver* motor DC dalam artikel ini adalah *driver* motor DC tipe *H-Bridge* menggunakan *power driver* berupa transistor. Rangkaian *driver* motor DC H-Bridge transistor ini dapat mengendalikan arah putaran motor DC dalam 2 arah dan dapat dikontrol dengan metode PWM (*Pulse Width Modulation*) maupun metode sinyal logika dasar TTL (*Driver*) dan (Low). Untuk pengendalian motor DC dengan metode PWM maka dengan rangkaian *driver* motor DC ini kecepatan putaran motor DC dapat dikendalikan dengan baik. Apabila menggunakan metode logika TTL 0 dan 1 maka rangkaian ini hanya dapat mengendalikan arah putaran motor DC saja dengan kecepatan putaran motor DC maksimum. Rangkaian *driver* motor DC H-Bridge ini menggunakan rangkaian jembatan transistor 4 unit dengan proteksi impuls tegangan induksi motor DC berupa dioda yang dipasang paralel dengan masing-masing transistor secara reverse bias.

Rangkaian *driver* motor DC secara detail dapat dilihat pada gambar berikut



**Gambar 2.17** Rangkaian *driver* motor dc

*# Halaman ini sengaja dikosongkan #*

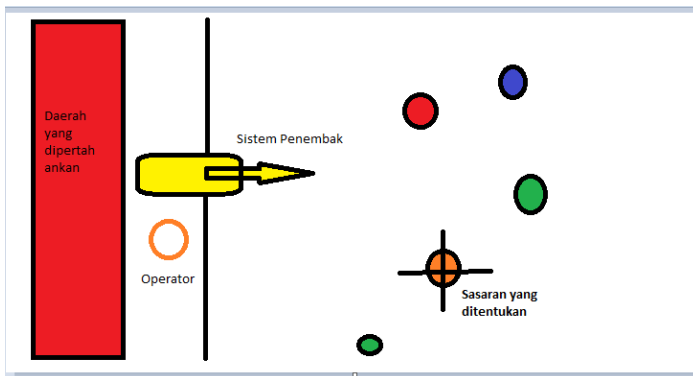
### BAB III

## PERANCANGAN SISTEM

Pada tugas akhir ini memanfaatkan teknologi *image processing* untuk mendapatkan target yang dituju lalu dilakukan penjejakan terhadapnya. Sistem penjejakan ini diterapkan pada daerah-daerah yang memerlukan pertahanan dari ancaman atau teror.

Pada awal sistem bekerja, operator akan menentukan sasaran yang sudah didapatkan dari kamera, sasaran yang perlu dijejak atau harus dilakukan penembakan terhadap sasaran tersebut. Setelah operator menentukan sasaran, maka sistem penjejak senjata otomatis akan melakukan tugas menjejak dan bisa pula melakukan penembakan.

Sistem ini diletakan di daerah pertahanan untuk melindungi daerah di belakangnya. Sistem pertahanan ini merupakan sistem pertahanan yang diletakan pada titik-titik tertentu saja. Sasaran utama pada sistem ini ialah manusia yang menjadi teror dan telah ditentukan sebagai target yang akan dijejak. Sistem ini memerlukan operator untuk memilih target atau sasaran mana yang perlu dijejak.



**Gambar 3.1** Sistem dengan pemilihan target

Pertama-tama, kamera yang diletakan pada sistem penembak akan mengambil gambar dan menampilkannya pada layar operator. Software pada sistem ini hanya akan mendeteksi keberadaan manusia karena hanya manusia yang menjadi fokus utama untuk penjejakan dan penembakan. Operator akan memilih manusia mana yang dapat dijejak dan ditembak.

Setelah operator memilih sasaran yang akan dijejek maka sistem penembak akan mengarahkan senjata ke arah sasaran dan akan terus mengikuti gerakan dari sasaran. Semua proses yang dilakukan operator berada pada processing unit, operator bisa mengatur sistem dari jarak aman dari ancaman atau teror. Diagram blok sistem sebagai berikut:



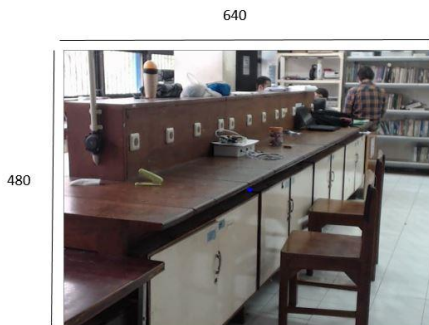
**Gambar 3.2** Diagram blok sistem

### 3.1 Perancangan *Software*

Pada tahap ini dilakukan pengolahan citra, dan membentuk kerja dari penjejakan serta penembakan. Pengolahan citra terdiri dari pengambilan citra, diubah menjadi *grayscale*, *optical flow*. Pengolahan citra dan penjejakan dilimpahkan ke *processing unit*.

#### 3.1.1 Pengambilan Citra

Tahap awal pada sistem yang direncanakan tidak lain adalah pengambilan citra oleh kamera. Pada tahap ini pengambilan citra diambil dengan ukuran 640x480.



**Gambar 3.3** Pengambilan citra 640x480



### 3.1.2 Grayscale

Setelah pengambilan citra selesai, tahap selanjutnya ada mengubah citra BGR menjadi citra Grayscale. Ini diperlukan dalam metode *optical flow* agar perhitungannya menjadi tidak terlalu kompleks seperti citra BGR yang memiliki 24 bit, untuk masing-masing warna *blue* 8 bit, *green* 8 bit, *red* 8 bit.

Dalam kasus cahaya yang ditransmisikan (misalnya, gambar di layar komputer), tingkat kecerahan merah (R), hijau (G) dan biru (B) komponen masing-masing diwakili sebagai angka dari desimal 0 sampai 255, atau biner 00000000 untuk 11111111. untuk setiap pixel dalam sebuah gambar grayscale merah-hijau-biru (RGB),  $R = G = B$ . ringan dari abu-abu yang berbanding lurus dengan jumlah mewakili tingkat kecerahan warna primer. Hitam diwakili oleh  $R = G = B = 0$  atau  $R = G = B = 00000000$ , dan putih diwakili oleh  $R = G = B = 255$  atau  $R = G = B = 11111111$ . Karena ada 8 bit dalam biner representasi dari tingkat abu-abu, metode pencitraan ini disebut 8-bit grayscale.



**Gambar 3.4** Hasil citra grayscale pada video capture

Berikut program yang digunakan untuk merubah citra RGB menjadi grayscale, pada QT opencv.

```
capwebcam.read(matOriginal);  
if (matOriginal.empty() == true) return;  
cv::cvtColor(matOriginal, matOriginal, CV_BGR2RGB);  
cv::cvtColor(matOriginal, img2, CV_BGR2GRAY);
```

keterangan:

capwebcam	: pengambilan citra dengan kamera
matOriginal	: sumber frame yang akan dikonversi
CV_BGR2RGB	: frame dikonversi menjadi citra RGB
Img2	: wadah peletakan hasil konversi
CV_BGR2GRAY	: perintah untuk mengkonversi RGB to GRAY

### 3.1.3 *Optical flow Lucas-Kanade*

Pada tugas akhir ini diterapkan metode *optical flow* Lucas-Kanade sebagai metode untuk menjejak target. Metode lukas kanade mengasumsikan bahwa aliran pada daerah sekitar suatu piksel biasanya konstan, lalu menyelesaikan dasar perhitungan *optical flow* untuk semua piksel di daerah sekitar, dari kriteria kuadrat terkecil.

Dalam *optical flow*, referensi yang telah dipilih akan dihitung intensitas *brightness* dari referensi tersebut. Dengan asumsi referensi piksel di lokasi (x,y,t) dengan intensitas  $I(x,y,t)$  melakukan perpindahan sebanyak  $\Delta x$ ,  $\Delta y$ , dan  $\Delta t$  pada frame berikutnya, ketetapan *brightness* dapat diberikan:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)$$

Sebelum senjata melakukan penjejakan terlebih dahulu operator memberikan titik atau *point* pada target yang dipilih untuk dijejak. Hanya satu target saja yang dijejak. Berikut pemberian *point* pada target yang telah di program.



**Gambar 3.5** Penandaan target yang dipilih operator

#### **3.1.4 Fuzzy Logic Controller**

Di dalam sistem ini kontroler yang digunakan adalah kontroler fuzzy. Kontroler ini digunakan untuk menentukan kecepatan dan arah gerak motor dc horizontal dan vertikal. Dalam pelacakan berdasarkan *image based visual servoing*, eror yang digunakan adalah selisih dari posisi piksel aktual sasaran terhadap posisi tengah frame dan eror inilah yang menjadi *input fuzzy logic controller* dan eror inilah yang menjadi acuan untuk seberapa besar kontrol yang diperlukan untuk mengunci dan menjejak sasaran. Keluaran dari *fuzzy logic controller* ini berupa nilai tegangan yang akan di proses lebih lanjut untuk mengatur *duty cycle* pada PWM.

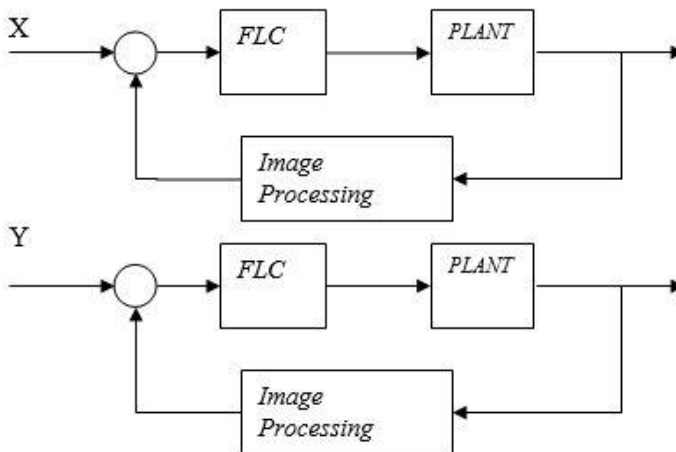
Ada tiga proses utama jika ingin mengimplementasikan fuzzy logic pada suatu perangkat, yaitu fuzzifikasi, evaluasi rule, dan defuzzifikasi.

1. *Fuzzification*, merupakan suatu proses untuk mengubah suatu masukan dari bentuk tegas (crisp) menjadi fuzzy yang biasanya disajikan dalam bentuk himpunan-himpunan fuzzy dengan suatu fungsi keanggotaannya masing-masing.
2. *Interference System* (Evaluasi Rule), merupakan sebagai acuan untuk menjelaskan hubungan antara variabel-variabel masukan

dan keluaran yang mana variabel yang diproses dan yang dihasilkan berbentuk fuzzy. Untuk menjelaskan hubungan antara masukan dan keluaran biasanya menggunakan “IF-THEN”.

3. *Defuzzification*, merupakan proses perubahan variabel berbentuk fuzzy tersebut menjadi data-data pasti (crisp) yang dapat dikirimkan ke peralatan pengendalian.

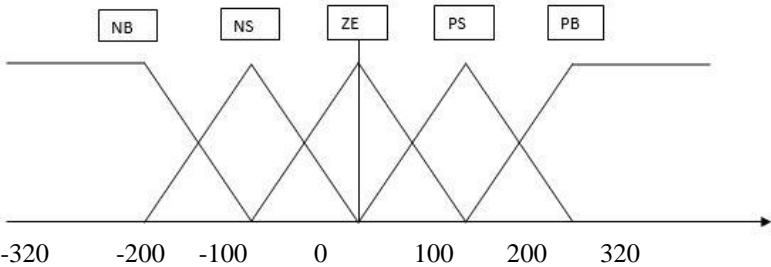
*Input* fuzzy berasal dari hasil image processing berupa koordinat x,y. Berikut merupakan bagan sistem penjejak:



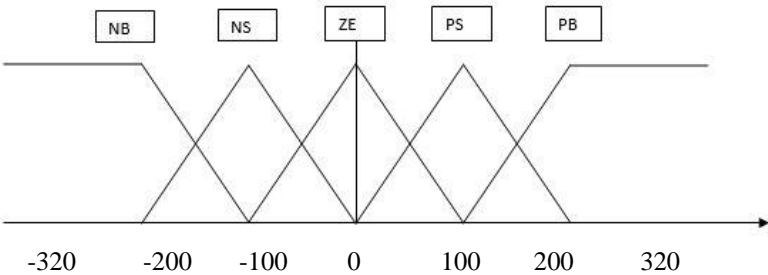
**Gambar 3.6** Diagram blok penjejak dengan fuzzy

Untuk *input* fuzzy berupa error dan delta error dari koordinat masing-masing.

Untuk fungsi keanggotaan *error* horizontal:



Untuk fungsi keanggotaan delta *error* horizontal:

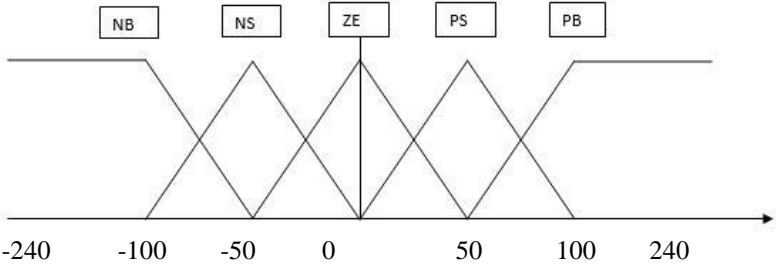


**Tabel 3.1** *Fuzzy Rule Base Horizontal*[8]

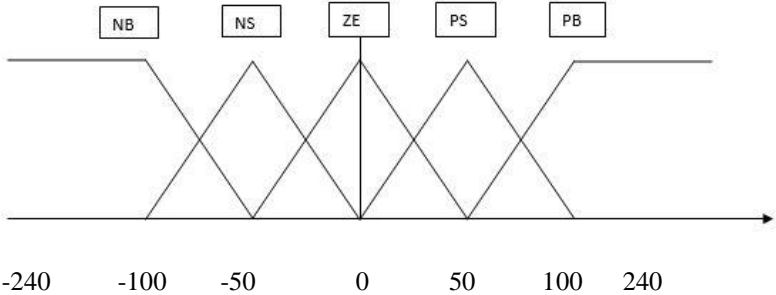
error Δerror \	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

NB = Negative Big  
 NS = Negative Small  
 ZE = Zero  
 PS = Positive Small  
 PB = Positive Big

Untuk fungsi keanggotaan *error* vertikal



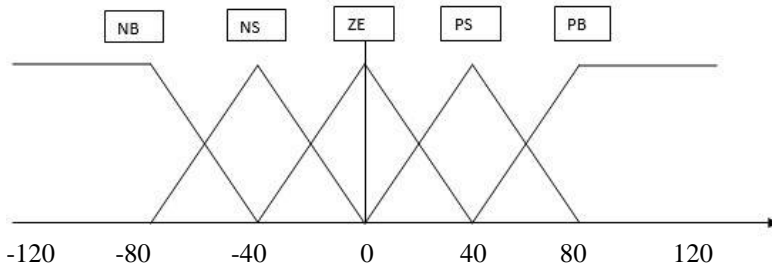
Untuk fungsi keanggotaan delta *error* vertikal



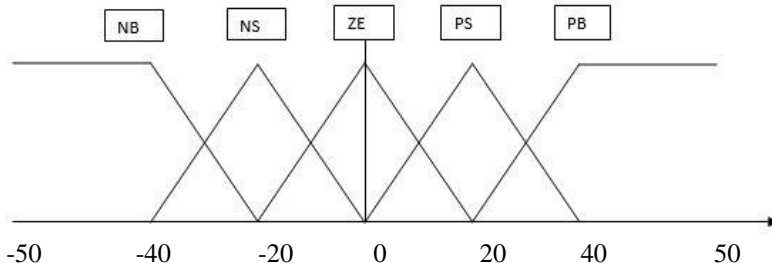
**Tabel 3.2** *Fuzzy Rule Base* Vertikal[8]

<div>error Δerror</div>	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NB	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PB	PB
PB	ZE	PS	PB	PB	PB

Fungsi keanggotaan untuk PWM horizontal:



Fungsi keanggotaan untuk PWM vertikal:



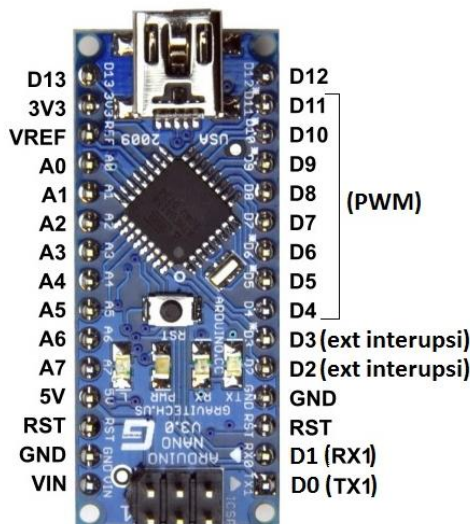
## 3.2 Perancangan *Hardware*

Pada bagian ini terdapat komponen-komponen perangkat keras yang digunakan pada sistem penjejak target otomatis

### 3.2.1 **Arduino Nano**

Pada tugas akhir ini digunakan mikrokontroler arduino nano untuk memberikan perintah gerak kiri-kanan ataupun atas-bawah. Mikrokontroler ini menerima nilai PWM dari komputer yang selanjutnya akan diberikan kepada *driver* motor. Arduino Nano adalah salah satu varian dari produk board mikrokontroler keluaran Arduino. Arduino Nano adalah board Arduino terkecil, menggunakan mikrokontroler Atmega 328 untuk Arduino Nano 3.x dan Atmega168 untuk Arduino Nano 2.x. Varian ini mempunyai rangkaian yang sama dengan jenis Arduino Duemilanove, tetapi dengan ukuran dan desain PCB yang berbeda. Arduino Nano tidak dilengkapi dengan soket catudaya, tetapi

terdapat pin untuk catu daya luar atau dapat menggunakan catu daya dari mini USB port. Arduino Nano didesain dan diproduksi oleh Gravitech.



**Gambar 3.7** Board arduino nano

Pada tugas akhir ini digunakan 2 *output* sebagai keluaran PWM horizontal dan PWM vertikal. Dan digunakan 4 *output* untuk mengatur gerak kiri-kanan dan atas-bawah. Berikut tabel lengkap port pada arduino nano yang digunakan untuk tugas akhir ini.

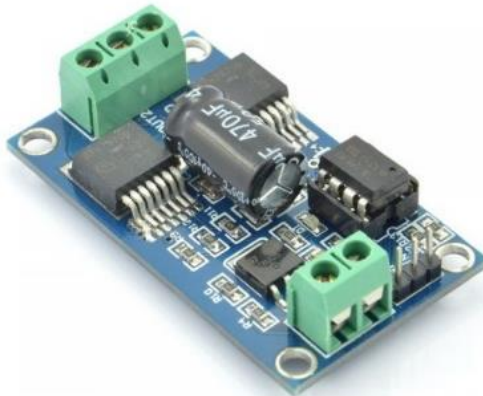


**Tabel 3.3** Penggunaan port arduino nano

Port	Fungsi
D3	Untuk keluaran PWM vertikal
D8	Untuk keluaran PWM horizontal
D4	Pin 1 untuk <i>driver</i> motor vertikal
D5	Pin 2 untuk <i>driver</i> motor vertikal
D6	Pin 1 untuk <i>driver</i> motor horizontal
D7	Pin 2 untuk <i>driver</i> motor horizontal
D1	TX pin transmit data
D0	RX pin penerima data dari komputer

### 3.2.2 *Driver motor*

*Driver Motor* disini menggunakan BTN7970. Dalam tugas akhir ini menggunakan 2 buah *driver* motor. 1 untuk motor penggerak vertikal dan 1 lagi untuk motor penggerak horizontal. Berikut gambaran dari *driver* motor yang digunakan:



**Gambar 3.8** *Driver* motor BTN7970

**Tabel 3.4** Tabel kebenaran *driver* motor Vertikal

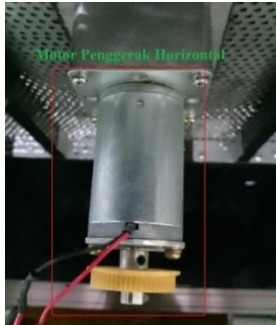
<i>Driver</i> Motor Vertikal		
In 1	In 2	Aksi pergerakan
0	0	Diam
0	1	Bawah
1	0	Atas
1	1	Diam

**Tabel 3.5** Tabel kebenaran *driver* motor Horizontal

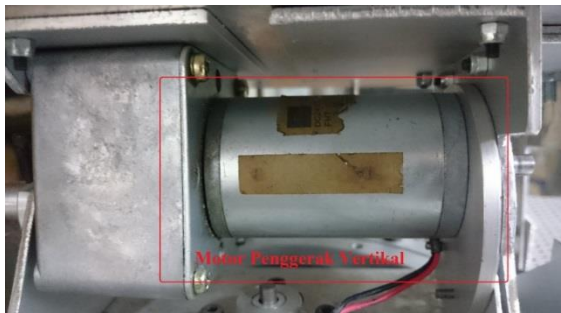
<i>Driver</i> Motor Horizontal		
In 1	In 2	Aksi pergerakan
0	0	Diam
0	1	Kiri
1	0	Kanan
1	1	Diam

### 3.2.3 Motor DC

Motor yang digunakan sistem penggerak horizontal dan vertikal pada tugas akhir ini adalah motor DC yang bisa bergerak pada 2 arah yang berlawanan. Arah pergerakan motor ditentukan oleh pemberian tegangan DC pada kedua *input* motor. Berikut gambaran motor DC yang digunakan pada tugas akhir ini beserta peletakannya.



**Gambar 3.9** Motor horizontal



**Gambar 3.10** Motor vertikal

### **3.2.4 Senjata**

Senjata yang digunakan pada Tugas Akhir ini adalah purwarupa SIG552



**Gambar 3.11** SIG552

### 3.2.5 Kamera

Kamera yang digunakan pada tugas akhir ini adalah Logitech C920



**Gambar 3.12** Kamera Logitech C920

Spesifikasi:

- Panggilan video Full HD 1080p (hingga 1920 x 1080 piksel) dengan versi terbaru Skype untuk Windows
- Panggilan video 720p HD (hingga 1280 x 720 piksel) dengan klien yang didukung
- Perekaman video Full HD (hingga 1920 x 1080 piksel) Kompresi video H.264
- Mikrofon dual stereo bawaan dengan pengurangan noise otomatis
- Koreksi pencahayaan-redup otomatis

### 3.2.6 Notebook

Pada tugas akhir ini notebook digunakan untuk proses pengolahan citra dan logika fuzzy. Spesifikasi dari *notebook* yang digunakan:

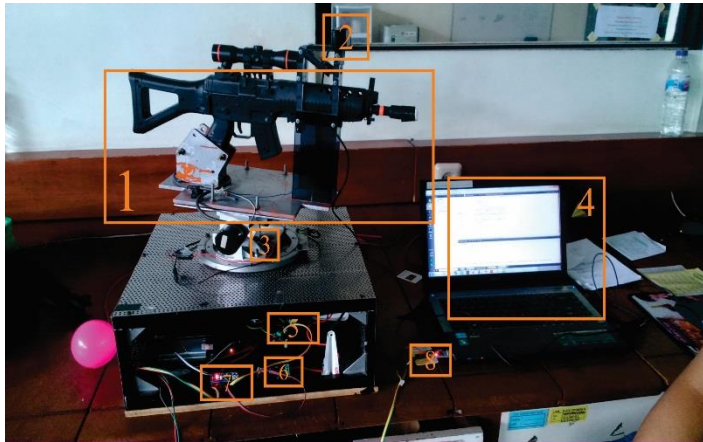
- *Manufacturer:* Acer
- *Processor:* Inter(R) Core(TM) i5--2430M CPU @ 2.40Ghz 2.40 Ghz
- *Installed memory:* (RAM) 4,00 GB (3,85 GB usable)
- *System type:* 64-bit Operating System

## BAB IV

### PENGUJIAN DAN ANALISIS

#### 4.1 Sistem Penjejak

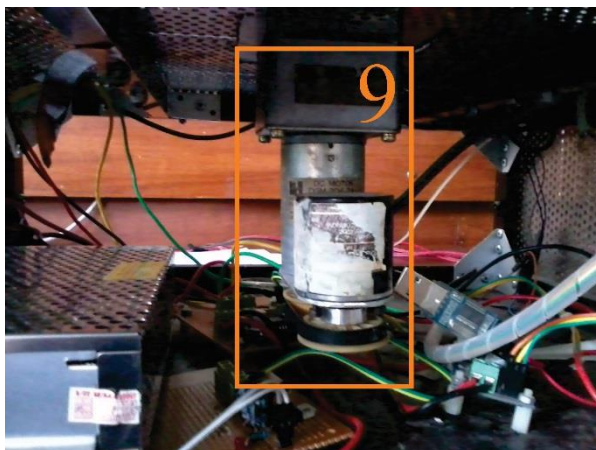
Berikut merupakan sistem penjejak yang telah dirancang pada tugas akhir ini yang memberikan letak *hardware* yang digunakan:



**Gambar 4.1** Sistem penjejak

Keterangan:

1. Senjata
2. Kamera
3. Motor DC vertikal
4. Notebook
5. Driver motor horizontal
6. Driver motor vertikal
7. Arduino nano
8. USB to TTL



**Gambar 4.2** Sistem penjejak

Keterangan :

9. Motor DC horizontal

## 4.2 Pengujian Motor

Pengujian ini dilakukan untuk mengetahui *duty cycle* minimum yang dapat digunakan untuk menggerakkan motor vertikal maupun motor horizontal.

**Tabel 4.1** Pengujian motor DC horizontal

Kondisi Beban	Massa beban	Duty cycle minimum	
		Dalam persen (%)	Dalam nilai 0-255
Tanpa beban	0 Kg	30%	77
SIG 552	0,5 Kg	35%	88

Dari data yang diperoleh pada pengujian di atas menjelaskan bahwa *duty cycle* minimum pada kondisi memiliki beban membutuhkan *duty cycle* yang lebih besar dari kondisi tanpa beban.

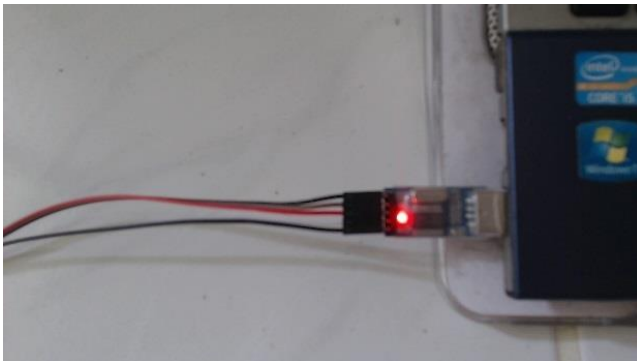
**Tabel 4.2** Pengujian motor DC vertikal

Kondisi beban	Massa beban	Duty cycle minimum	
		Dalam persen (%)	Dalam nilai 0-255
Tanpa beban	0 Kg	15%	38
SIG 552	0,5 Kg	17%	44

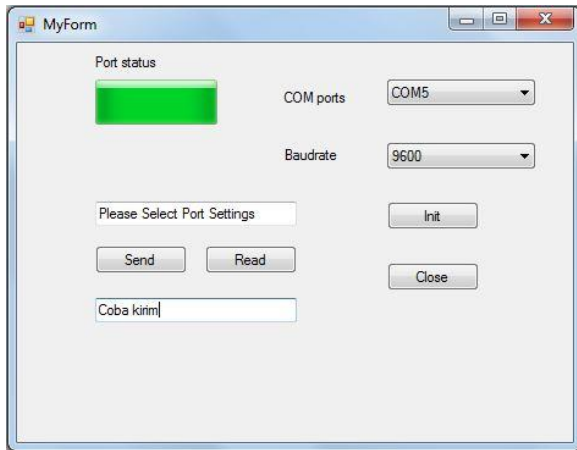
Dari data yang diperoleh pada pengujian di atas menjelaskan bahwa *duty cycle* minimum pada kondisi memiliki beban membutuhkan *duty cycle* yang lebih besar dari kondisi tanpa beban.

### 4.3 Pengujian Serial Port USB to TTL

USB to TTL ini digunakan untuk komunikasi serial notebook dengan *microprocessor*



**Gambar 4.3** USB to TTL



**Gambar 4.4** Pengecekan port USB to TTL

Untuk pengecekan COM port yang tersedia jika di plug oleh USB to TTL juga dapat dilihat dari *Device Manager* yang terdapat pada menu sistem komputer.

#### 4.4 Pengujian *Optial Flow* Pada Siang Hari

Dengan metode *Optical flow lukas kanade*, dilakukan penjejukan terhadap objek manusia:



**Gambar 4.5** Uji *optical flow*



Dilakukan pengujian *optical flow* pada target manusia, titik untuk penjejakan diletakkan pada dada kanan target yang selanjutnya akan melakukan pergerakan untuk melihat kemampuan jejak. Target bergerak maju lalu bergerak ke arah kiri seperti yang terlihat pada gambar 4.6., target bergerak dari posisi semula sampai posisi di gambar 4.6 dengan berjalan



**Gambar 4.6** Uji *Optical flow*

Hasil yang diperoleh dari penjejakan di atas menunjukkan *optical flow* bekerja dengan cukup baik pada kondisi cukup pencahayaan.

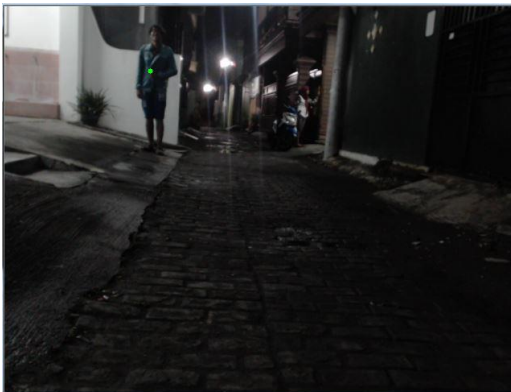


**Gambar 4.7** Uji *optical flow*

Pengujian selanjutnya target bergerak ke kanan lalu membalikkan badan, sehingga titik yang terletak pada dada kanan berpindah ke bahu kiri. Hasil tersebut disebabkan oleh hilangnya *feature image* yang telah dipilih sebelumnya pada dada kanan, maka titik tersebut akan mencari nilai piksel sekitar yang mirip dengan dada kanan, terlihat titik masih terdapat pada target akan tetapi berpindah tempat.

#### 4.5 Pengujian *Optical Flow* Pada Malam Hari

Dengan metode yang telah dijelaskan sebelumnya, pengujian untuk penjejakan target dilakukan dalam kondisi tidak adanya cahaya matahari, hanya saja mendapatkan cahaya dari lampu, hasil pengujian penjejakan target sebagai berikut:



**Gambar 4.8** Uji coba *optical flow* malam hari

Pengujian ini dilakukan untuk mengetahui kehandalan metode *optical flow* pada kondisi tidak cukup mendapat cahaya. Target yang telah dipilih adalah manusia, titik diletakkan pada bagian perut, selanjutnya target bergerak ke kiri seperti yang terlihat pada gambar 4.9





**Gambar 4.9** Uji coba *optical flow* malam hari

Target di atas memiliki warna *brighness* yang hampir sama dengan *background*, *pointer* mudah lepas dari target sehingga hasil penjejakan tidak cukup baik pada kondisi kurangnya pencahayaan.



#### **4.6 Pengujian Performa Sistem**

Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan dan error yang dimiliki oleh sistem dalam melakukan penjejakan, serta waktu yang dibutuhkan senjata untuk segera menjejak target yang telah dipilih oleh operator. Target bergerak horizontal dari tengah frame ke kiri, tengah frame ke kanan, dari kiri ke kanan, dan dari kanan ke kiri, sedangkan pergerakan vertikal target bergerak dari tengah frame ke atas, tengah frame ke bawah, bawah ke atas, dan atas ke bawah. Berikut ilustrasi gerak target untuk uji coba performa pada tabel 4.3 dan pada tabel 4.6. Posisi titik tengah pada *frame* diasumsikan berada pada piksel 280 sampai 360 untuk horizontal, sedangkan asumsi titik tengah vertikal berada pada piksel 220 sampai 280.


**Tabel 4.3** Ilustrasi pergerakan target untuk uji coba

No	Gambar	Keterangan
1		<p>Penempatan posisi titik di tengah</p>
2		<p>Target bergerak dari tengah ke kanan</p>

Lanjutan **Tabel 4.3** Ilustrasi pergerakan target untuk uji coba

No	Gambar	Keterangan
3		Target bergerak dari kanan ke kiri
4		Target bergerak dari kiri ke kanan

Lanjutan **Tabel 4.3** Ilustrasi pergerakan target untuk uji coba

No	Gambar	Keterangan
5		Target bergerak dari tengah ke kiri

**Tabel 4.4** Pengujian performa sistem gerak horizontal


No	Arah gerak	Piksel			Error	Time
		Initial point	Set point	Posisi akhir jejak		
1	Tengah ke kiri	329	320	359	12,18%	4s
2	Tengah ke kiri	335	320	334	4,37%	3s
3	Tengah ke kiri	326	320	345	7,81%	3s
4	Tengah ke kiri	339	320	334	4,37%	3s
5	Tengah ke kiri	356	320	380	18,75%	4s
6	Tengah ke kiri	338	320	359	12,18%	3s
7	Tengah ke kiri	342	320	312	2,5%	2s
8	Tengah ke kiri	334	320	302	5,62%	2s
9	Tengah ke kiri	360	320	361	12,81%	4s
10	Tengah ke kiri	351	320	344	7,5%	3s
11	Kiri ke kanan	115	320	266	16,87%	2s
12	Kiri ke kanan	123	320	343	7,18%	2s
13	Kiri ke kanan	35	320	302	5,62%	3s

Lanjutan **Tabel 4.4** Pengujian performa sistem gerak horizontal

No	Arah gerak	Piksel			Error	Time
		<i>Initial point</i>	<i>Set point</i>	Posisi jejak akhir		
14	Kiri ke kanan	70	320	310	3,12%	3s
15	Kiri ke kanan	88	320	278	13,12%	2s
16	Kiri ke kanan	113	320	352	10%	2s
17	Kiri ke kanan	129	320	321	0,31%	2s
18	Kiri ke kanan	86	320	352	10%	3s
19	Kiri ke kanan	92	320	266	16,87%	3s
20	Kiri ke kanan	107	320	343	7,18%	2s
21	Tengah ke kanan	336	320	259	19,06%	3s
22	Tengah ke kanan	351	320	274	14,37%	3s
23	Tengah ke kanan	323	320	288	10%	3s
24	Tengah ke kanan	332	320	355	10,93%	2s
25	Tengah ke kanan	340	320	247	22,8%	3s
26	Tengah ke kanan	331	320	274	14,37%	3s
27	Tengah ke kanan	346	320	345	7,81%	4s
28	Tengan ke kanan	340	320	330	3,125%	2s
29	Tengah ke kanan	330	320	355	10,93%	2s
30	Tengah ke kanan	327	320	341	6,56%	2s
31	Kanan ke kiri	524	320	325	1,56%	2s
32	Kanan ke kiri	569	320	349	9,06%	3s
33	Kanan ke kiri	532	320	336	5%	3s
34	Kanan ke kiri	573	320	351	9,68%	4s
35	Kanan ke kiri	578	320	278	13,12%	3s
36	Kanan ke kiri	552	320	385	20,31%	3s
37	Kanan ke kiri	562	320	300	6,25%	2s
38	Kanan ke kiri	543	320	318	0,62%	2s
39	Kanan ke kiri	568	320	262	18,25%	3s
40	Kanan ke kiri	541	320	285	10,93%	2s



Dari hasil pengujian di atas sistem bekerja belum cukup baik karena masih memiliki persentase kesalahan yang tertinggi pada 22,8% dengan rata-rata kesalahan gerak tengah ke kiri sebesar 6,716%, kiri ke kanan sebesar 9,027%, tengah ke kanan sebesar 17,805%, dan kanan ke kiri sebesar 9,478%. Hasil uji di atas dipengaruhi oleh kecepatan gerak target, target berupa manusia. Apabila manusia bergerak maka dapat membuat kesalahan menjadi tinggi dan mampu menghilangkan pejejakan pada manusia yang terpilih menjadi target. Cahaya yang kurang terang juga menyebabkan proses pengambilan gambar lambat sehingga menyebabkan gerak motor mengalami kelebihan dan pergeseran *image* mengalami *blur*. Sedangkan untuk perbedaan warna antara sasaran dengan *background* memiliki pengaruh yang besar juga. Karena metode ini melihat pergeseran *flow* dari warna pada piksel, jika terdapat gradasi warna yang rendah maka posisi sasaran bisa berubah dan dapat lepas dari sasaran. Selanjutnya dilakukan pengujian terhadap gerak vertikal dengan target berupa bola.

**Tabel 4.5** Ilustrasi pergerakan bola



No	Gambar	Keterangan
1		Bola tertelak di tengah



Lanjutan **Tabel 4.5** Ilustrasi pergerakan bola

No	Gambar	Keterangan
2		<p>Bola bergerak dari tengah ke bawah</p>
3		<p>Bola bergerak dari bawah ke atas</p>

Lanjutan **Tabel 4.5** Ilustrasi pergerakan bola

No	Gambar	Keterangan
4		<p>Bola bergerak dari atas ke bawah</p>
5		<p>Bola bergerak dari tengah ke atas</p>

**Tabel 4.6** Pengujian performa sistem gerak vertikal

No	Arah gerak	Piksel			Error	Time
		<i>Initial point</i>	<i>Set point</i>	Posisi jejak akhir		
1	Tengah ke atas	243	240	221	7.91%	3s
2	Tengah ke atas	231	240	209	12.91%	4s
3	Tengah ke atas	222	240	207	13.75%	4s
4	Tengah ke atas	246	240	261	8.75%	3s
5	Tengah ke atas	251	240	236	1.66%	4s
6	Tengah ke atas	242	240	251	4.58%	5s
7	Tengah ke atas	238	240	210	12.5%	4s
8	Tengah ke atas	279	240	234	1.87%	4s
9	Tengah ke atas	265	240	207	13.75%	4s
10	Tengah ke atas	233	240	221	7.91%	3s
11	Atas ke bawah	121	240	217	9.58%	4s
12	Atas ke bawah	132	240	211	12.08%	3s
13	Atas ke bawah	97	240	214	10.83%	5s
14	Atas ke bawah	81	240	208	13.33%	5s
15	Atas ke bawah	138	240	208	13.33%	3s
16	Atas ke bawah	122	240	215	10.41%	4s
17	Atas ke bawah	168	240	217	9.58%	3s
18	Atas ke bawah	133	240	220	8.33%	3s
19	Atas ke bawah	142	240	216	10%	3s
20	Atas ke bawah	124	240	215	10.41%	3s
21	Tengah ke bawah	233	240	206	14.16%	4s
22	Tengah ke bawah	241	240	207	13.75%	4s
23	Tengah ke bawah	253	240	202	15.83%	5s
24	Tengah ke bawah	238	240	204	15%	4s
25	Tengah ke bawah	247	240	205	14.58%	4s
26	Tengah ke bawah	255	240	206	14.16%	5s

Lanjutan **Tabel 4.6** Pengujian performa sistem gerak vertikal

27	Tengah ke bawah	236	240	202	15.83%	4s
28	Tengah ke bawah	229	240	202	15.83%	4s
29	Tengah ke bawah	239	240	204	15%	3s
30	Tengah ke bawah	245	240	203	15.41%	4s
31	Bawah ke atas	363	240	215	10.41%	4s
32	Bawah ke atas	340	240	207	13.75%	3s
33	Bawah ke atas	357	240	209	12.91%	3s
34	Bawah ke atas	379	240	222	7.5%	5s
35	Bawah ke atas	348	240	204	15%	4s
36	Bawah ke atas	338	240	225	6.25%	5s
37	Bawah ke atas	369	240	206	14.16%	5s
38	Bawah ke atas	398	240	209	12.91%	5s
39	Bawah ke atas	363	240	210	12.5%	4s
40	Bawah ke atas	479	240	223	7.08%	5s

Dari hasil pengujian di atas sistem bekerja belum cukup baik karena masih memiliki persentase kelasahan yang tertinggi pada 15,83% dengan rata-rata kesalahan gerak tengah ke atas sebesar 8,559%, atas ke bawah sebesar 10,788%, tengah ke bawah sebesar 14,955%, dan bawah ke atas sebesar 11,247% .Hasil uji di atas dipengaruhi oleh kecepatan gerak target, target berupa bola. Bola diletakkan di tangan, apabila gerakan bola terlalu cepat, maka point atau titik yang telah dijejak dapat hilang dari bola terpilih.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Kesimpulan yang diperoleh pada tugas akhir, yakni: .

1. Massa dari senjata mempengaruhi duty cycle minimum yang dibutuhkan motor untuk dapat bergerak.
2. Perancangan fungsi keanggotaan fuzzy didapat dari karakteristik motor, sehingga sistem mampu bekerja sesuai dengan kemampuan motor
3. Penjejakan target yang bergerak horizontal dengan metode *optical flow* dan proses fuzzy memiliki kesalahan kurang dari 23% dengan rata-rata kesalahan gerak tengah ke kiri sebesar 6,716%, kiri ke kanan sebesar 9,027%, tengah ke kanan sebesar 17,805%, dan kanan ke kiri sebesar 9,478%
4. Penjejakan target yang bergerak vertikal dengan metode *optical flow* dan proses fuzzy memiliki kesalahan kurang dari 16% dengan rata-rata kesalahan gerak tengah ke atas sebesar 8,559%, atas ke bawah sebesar 10,788%, tengah ke bawah sebesar 14,955%, dan bawah ke atas sebesar 11,247%
5. Penjejakan target dengan metode *optical flow* bekerja buruk pada target yang bergerak cepat dan memiliki gradasi warna yang mirip dengan *background*

#### **5.2 Saran**

Beberapa saran dari penulis untuk kedepannya:

1. Kamera yang digunakan diharapkan memiliki *Frame Per Second* (FPS) yang lebih tinggi dari yang telah dipakai
2. Dapat dicari metode penjejakan untuk gerak target yang cepat dan tidak terpengaruh halangan.
3. Kamera yang digunakan memiliki mode *night vision* sehingga bisa dicoba untuk kondisi gelap.

*#Halaman ini sengaja dikosongkan#*

## DAFTAR PUSTAKA

- [1] Putra, Darma, "Pengolahan Citra Digital", Yogyakarta, Penerbit ANDI, 2010.
- [2] Achmad Rizal, "Pengolahan Citra", <URL: <http://achmadrizal.staff.telkomuniversity.ac.id/2014/06/19/pengolahan-citra/>>, Juni, 2014.
- [3] Viola P. & Jones M (2001) Rapid Object Detection using a Boosted Cascade of Simple Features Computer Vision and Pattern Recognition Proceedings of the 2001 IEEE Computer Society Conference vol. 1: I-511 – I-518
- [4] Bouget, J. Y., "Pyramidal Implementation of The Lucas-Kanade Feature Tracker", Intel Cooperation, 2000
- [5] Amar "Fuzzy Logic"  
<https://amarnotes.wordpress.com/2013/10/14/apa-itu-fuzzy-logic/>
- [6] Anonymus 31 Oktober  
<http://caramake.blogspot.co.id/2016/10/cara-menggunakan-usb-to-ttl-adapter-usb.html>
- [7] Bradski, Gary dan Kaebler, Adrian. "*Learning OpenCV Computer Vision with the OpenCV Library*", O' REILLY. 2008.
- [8] Eko Supriyanto, Yeo Kee Jiar, Tan Yong Oon, Tan Meng Kuan "Facial Tracking based Camera Motion Control System". University Teknologi Malaysia. 2010

*#Halaman ini sengaja dikosongkan#*



## LAMPIRAN

Berikut merupakan program yang telah dibuat:

Deklarasi dan fungsi di header:

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QMessageBox>
#include <QtCore>
#include <QSerialPort>
#include <QSerialPortInfo>

#include <vector>
#include <string>
#include <algorithm>
#include <stdio.h>
#include <iostream>
#include <iterator>
#include <stdlib.h>
#include <ctype.h>

#include <opencv2/core/core.hpp>
#include <opencv2/opencv2/drivergui/drivergui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/calib3d/calib3d.hpp>
#include <opencv2/contrib/contrib.hpp>
#include "opencv2/video/tracking.hpp"
namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT
```

```

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

private slots:
    void on_btnPauseOrResume_clicked();
    void kirim (QString);

private:
    Ui::MainWindow *ui;
    cv::VideoCapture capwebcam;
    QPoint camera_pos;

    cv::Mat matOriginal;//img1
    cv::Mat img2; //imgG
    cv::Mat previmg2;
    QImage matToQImage(cv::Mat mat);
    int error0x;
    int error0y;
    int inputx ;
    int inputy ;
    int errorx ;
    int errory ;

    int delta_errorx;
    int delta_errory;

    std::vector<cv::Vec2f> vecCircles;
    std::vector<cv::Vec2f>::iterator itrCircles;

    QImage qimgOriginal;
    void LucasKanade(cv::Mat gray, cv::Mat prevGray);

```

```

float fuzzyHor(int ErrorX, int DeltaX);
float fuzzyVer(int ErrorY, int DeltaY);
float min(float A1, float B1);
float max(float A1, float B1);
float segitiga(float x,float g,float h);
float trapesium(float x,float c,float d,float e,float f);

//variable global untuk Lucas Kanade
cv::Point2f point;
QTimer* tmrTimer;
//serial

QSerialPort *SerialCom;

static const quint16 vendor_id = 1659;// usbToTTL_biasa: 1659;
CP2102: 4292 CH340: 6790
static const quint16 product_id= 8963;//usbToTTL_biasa:8963;
CP2102:60000; CH340: 29987
QString serial_port_name;
bool serial_is_available;
int pwmHor;
int pwmVer;
int waw;
int woy;

public slots:
void processFrameAndUpdateGUI();
void receiveMousePosition(QPoint& pos);//slot menerima QPoint
mouse
void showMousePosition(QPoint &pos);

};

#endif // MAINWINDOW_H

```

Program utama:

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QDebug>
#include <math.h>

#define PI acos(-1)

#include <QtCore>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    capwebcam.open(0);

    connect(ui->Display, SIGNAL(sendPosition(QPoint&)), this,
        SLOT(receiveMousePosition(QPoint&)));
    connect(ui->Display, SIGNAL(sendMouse(QPoint&)), this,
        SLOT(showMousePosition(QPoint&)));
    if (capwebcam.isOpened() == false){
        ui -> txtXY -> appendPlainText("Error: webcam not accessed
successfully ");
        return;
    }
    tmrTimer = new QTimer(this);
    connect(tmrTimer, SIGNAL(timeout()), this,
        SLOT(processFrameAndUpdateGUI()));
    tmrTimer -> start (200);
    camera_pos.setX(320);
    camera_pos.setY(240);
    error0x =0;
    error0y =0;

    point=cv::Point2f((float)camera_pos.x(), (float)camera_pos.y());
```

```

//setting serial
SerialCom = new QSerialPort;
serial_port_name = "COM5";
serial_is_available = false;

qDebug() << "Number of serial port: "<<
QSerialPortInfo::availablePorts().length();
foreach(const QSerialPortInfo &serialPortInfo,
QSerialPortInfo::availablePorts()){
    if(serialPortInfo.hasVendorIdentifier()){
        qDebug() << "Vendor ID: " << serialPortInfo.vendorIdentifier();
    }
    if(serialPortInfo.hasProductIdentifier()){
        qDebug() << "Product ID: " <<
serialPortInfo.productIdentifier();
    }
}

foreach(const QSerialPortInfo &serialport,
QSerialPortInfo::availablePorts()){
    if(serialport.hasVendorIdentifier() &&
serialport.hasProductIdentifier()){
        if(serialport.vendorIdentifier()==vendor_id &&
serialport.productIdentifier()==product_id){
            serial_port_name = serialport.portName();
            serial_is_available = true;
        }
    }
}

if(serial_is_available){
    SerialCom->setPortName(serial_port_name);
    SerialCom->open(QSerialPort::WriteOnly);
    SerialCom->setBaudRate(QSerialPort::Baud9600);
    SerialCom->setDataBits(QSerialPort::Data8);
    SerialCom->setParity(QSerialPort::NoParity);
    SerialCom->setStopBits(QSerialPort::OneStop);
    SerialCom->setFlowControl(QSerialPort::NoFlowControl);
}

```

```

    }
    else{
        QMessageBox::warning(this, "Port Error", "Couldn't find the serial
port");
    }

}

MainWindow::~MainWindow()
{
    if(SerialCom->isOpen()){
        SerialCom->close();
    }
    delete ui;
}
/////////////////////////////////////////////////////////////////
/////
void MainWindow:: kirim(QString command){
    if(SerialCom->isWritable()){
        SerialCom->write(command.toStdString().c_str());
    }else{
        qDebug() << "Tidak terkirim";
    }
}

/////////////////////////////////////////////////////////////////
/////
void MainWindow::processFrameAndUpdateGUI(){
    capwebcam.read(matOriginal);
    if (matOriginal.empty() == true) return;
    cv::cvtColor(matOriginal, matOriginal, CV_BGR2RGB);
    cv::flip(matOriginal, matOriginal, 1);
    cv::cvtColor(matOriginal, img2, CV_BGR2GRAY);
    if(previmg2.empty()) img2.copyTo(previmg2);
    LucasKanade(img2, previmg2);
    if(point.x < 0){
        point.x = 0;
    }
}

```

```

else if(point.x >= 640){
    point.x = 639;
}
if(point.y < 0){
    point.y = 0;
}
else if(point.y >= 480){
    point.y = 479;
}

cv::swap(previmg2, img2);
//cv::Point(camera_pos.x()-5, camera_pos.y()-5),
cv::Point(camera_pos.x()+5, camera_pos.y()+5), cv::Scalar(255, 0, 0),
2);
cv::circle(matOriginal, point,3, cv::Scalar(0, 255, 0), -1,8);

// camera_pos.setX((int)point.x);
// camera_pos.setY((int)point.y);

int inputx = (int)point.x;
int inputy = (int)point.y;
ui->label->setText("x: "+QString::number(inputx)+", y:
"+QString::number(inputy));
int errorx = inputx-320;
int errory = inputy-240;
ui->ErrorX->setText(QString::number(errorx));
ui->ErrorY->setText(QString::number(errory));
int delta_errorx = errorx-error0x;
int delta_errory = errory-error0y;

//    if (delta_errory >100){
//        delta_errory = 0;
//    }
fuzzyHor (errorx, delta_errorx);
fuzzyVer (errorx, delta_errory);
if ((error0y >-30) || (error0y <30)){pwmVer=0;}
fuzzyHor(errorx, delta_errorx);
fuzzyVer(errory, delta_errory);

```

```

MainWindow::kirim(QString("v%1").arg(pwmVer));
qDebug() << pwmVer;
MainWindow::kirim(QString("h%1").arg(pwmHor));
qDebug() << pwmHor;

QImage qImg1 = matToQImage(matOriginal);

ui->Display->setPixmap(QPixmap::fromImage(qImg1));
//ui->Display->setPixmap(QPixmap::fromImage(qimgOriginal));

}
//////////
//////////
void MainWindow::receiveMousePosition(QPoint &pos)
{
    camera_pos = pos;
    point=cv::Point2f((float)pos.x(), (float)pos.y());
}

//////////
//////////
void MainWindow::showMousePosition(QPoint &pos)
{
    camera_pos = pos;
    ui->label->setText("x: "+QString::number(pos.x())+", y:
"+QString::number(pos.y()));
}
//////////
//////////
QImage MainWindow::matToQImage(cv::Mat mat)
{
    if(mat.channels() ==1)
        return QImage((uchar*)mat.data, mat.cols, mat.rows, mat.step,
QImage::Format_Indexed8);
    else if(mat.channels() == 3)
        return QImage((uchar*)mat.data, mat.cols, mat.rows, mat.step,
QImage::Format_RGB888);
}

```



```

else
    return QImage();
}
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
void MainWindow::on_btnPauseOrResume_clicked()
{
    if (tmrTimer->isActive() == true){
        tmrTimer->stop();
        ui->btnPauseOrResume->setText("Resume");}
    else{
        tmrTimer->start(20);
        ui->btnPauseOrResume->setText("Pause");
    }

}
/////////////////////////////////////////////////////////////////
/////////////////////////////////////////////////////////////////
void MainWindow::LucasKanade(cv::Mat gray, cv::Mat prevGray)
{
    cv::TermCriteria
termcrit(CV_TERMCRIT_ITER|CV_TERMCRIT_EPS, 20, 0.03);
    cv::Size winSize(31,31);

    std::vector<cv::Point2f> points[2];
    points[0].push_back(point);

    std::vector<uchar> status;
    std::vector<float> err;
    calcOpticalFlowPyrLK(prevGray, gray, points[0], points[1], status,
err, winSize, 3, termcrit, 0, 0.001);
    point=points[1][0];
}
float MainWindow::segitiga(float x,float g,float h){
    float y=0;
    if ((x<=(g-h/2)) || ((x>= (g+h/2)) && h>0))
        y = 0;
    else if ((x > (g-h/2)) && (x < (g+h/2)) && h>0)
        y = 1 -2*abs(x-g) /h;
}

```

```

    else if (x==g && h==0)
        y=1;

    return y;
}

float MainWindow::trapesium(float x,float c,float d,float e,float f){
    float y=0;
    if (((x<(c-d)) && d>0) || ((x>(e+f)) && f>0))
        y=0;
    else if (x<c)
        y=(x-c+d)/d;
    else if ((x<=e || (x==c && d==0)|| (x==e && f==0)))
        y=1;
    else if (x<= (e+f))
        y=(e+f-x)/f;
    return y;
}

float MainWindow::min(float A1, float B1)
{if (A1>B1){ return B1;}
else
return A1;
}

float MainWindow::max(float A1, float B1)
{if (A1>B1){ return A1;}
else
return B1;
}

float MainWindow::fuzzyHor(int ErrorX, int DeltaX){
    //Fungsi keanggotaan
    //ErrorX

    float NB,NS,ZE,PS,PB,NB1,NS1,ZE1, PS1,
PB1,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,
x19,x20,x21,x22,x23,x24,x25;
    NB = trapesium (ErrorX, -320,0 ,-200 ,100 );
    NS = segitiga (ErrorX, -100,200 );
    ZE = segitiga (ErrorX, 0,200 );
    PS = segitiga (ErrorX, 100, 200);
    PB = trapesium (ErrorX, 200,100 , 320 ,0 );

```

```
//DeltaX
NB1 = trapesium (DeltaX,-320 , 0, -200,100 );
NS1 = segitiga (DeltaX,-100 ,200 );
ZE1 = segitiga (DeltaX, 0,200 );
PS1 = segitiga (DeltaX, 100,200 );
PB1 = trapesium (DeltaX, 200, 100, 320,0 );
```

```
//Rule
```

```
//NB
```

```
x1 = min (NB, NB1);
x2 = min (NB, NS1);
x3 = min (NB, ZE1);
x4 = min (NS, NB1);
x5 = min (NS, NS1);
x6 = min (ZE, NB1);
```

```
//NS
```

```
x7 = min (NB, PS1);
x8 = min (NS, ZE1);
x9 = min (ZE, NS1);
x10 = min (PS, NB1);
```

```
//ZE
```

```
x11 = min (NB, PB1);
x12 = min (NS, PS1);
x13 = min (ZE, ZE1);
x14 = min (PS, NS1);
x15 = min (PB, NB1);
```

```
//PS
```

```
x16 = min (NS, PB1);
x17 = min (ZE, PS1);
x18 = min (PS, ZE1);
x19 = min (PB, NS1);
```

```
//PB
```

```
x20 = min (ZE, PB1);
x21 = min (PS, PS1);
x22 = min (PS, PB1);
x23 = min (PB, ZE1);
x24 = min (PB, PS1);
x25 = min (PB, PB1);
float y1,y2,y3,y4,y5;
```



```

        while (i<40){yo[i] = max (min (y4, segitiga (xo[1],40,80)), min
(y5, trapesium(xo[i],80,40,120,0)));
        i=i+1;}

        while (i<47){yo[i] = min (y5, trapesium(xo[i],80,40,120,0));
        i=i+1;}
        float numerator = 0;
        float denominator = 0;

        for (int i=0;i<48;i++){
            denominator = denominator + yo[i];
            numerator = numerator + yo[i]*xo[i];
        }
        int z= numerator / denominator;
        ui->PWMhor->setText(QString::number(z));
        pwmHor = z;
        return z;

        error0x = errorx;
    }
    float MainWindow::fuzzyVer(int ErrorY, int DeltaY){
        //Fungsi keanggotaan
        //ErrorY

        float NB,NS,ZE,PS,PB,NB1,NS1,ZE1, PS1,
        PB1,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,
        x19,x20,x21,x22,x23,x24,x25;
        NB = trapesium (ErrorY, -240,0 ,-125 ,75 );
        NS = segitiga (ErrorY, -50,125 );
        ZE = segitiga (ErrorY, 0,70 );
        PS = segitiga (ErrorY, 50, 125);
        PB = trapesium (ErrorY, 125,75 ,240 ,0 );
        //DeltaY
        NB1 = trapesium (DeltaY,-240 , 0, -125,75 );
        NS1 = segitiga (DeltaY,-50 ,125 );
        ZE1 = segitiga (DeltaY, 0,70 );
        PS1 = segitiga (DeltaY, 50,125 );
        PB1 = trapesium (DeltaY, 125, 75, 240,0 );

```

```

//Rule
//NB
x1 = min (NB, NB1);
x2 = min (NB, NS1);
x3 = min (NB, ZE1);
x4 = min (NS, NB1);
x5 = min (NS, NS1);
x6 = min (ZE, NB1);
//NS
x7 = min (NB, PS1);
x8 = min (NS, ZE1);
x9 = min (ZE, NS1);
x10 = min (PS, NB1);
//ZE
x11 = min (NB, PB1);
x12 = min (NS, PS1);
x13 = min (ZE, ZE1);
x14 = min (PS, NS1);
x15 = min (PB, NB1);
//PS
x16 = min (NS, PB1);
x17 = min (ZE, PS1);
x18 = min (PS, ZE1);
x19 = min (PB, NS1);
//PB
x20 = min (ZE, PB1);
x21 = min (PS, PS1);
x22 = min (PS, PB1);
x23 = min (PB, ZE1);
x24 = min (PB, PS1);
x25 = min (PB, PB1);
float y1,y2,y3,y4,y5;
float j,k,l,m,n,o,p,q,r,s,t,u,v,w,woy;
j = max (x1,x2);k = max (x4,x5);l = max (x5,x6);m = max (k,l);
y1 = max (j,m);
n = max (x7,x8);o=max(x9,x10);
y2 = max (n,o);
p= max (x11,x12);q=max(x13,x14); r = max (p,q);
y3 = max (r,x15);

```

```

s = max (x16,x17); t = max(x18,x19);
y4 = max (s,t);
u= max (x20,x21);v= max(x22,x23); w=max(x24,x25);woy
=max(v,w);
y5 = max (u,woy);

```

```

float xo[19] = {-45,-40,-35,-30,-25,-20,-15,-10,-
5,0,5,10,15,20,25,30,35,40,45};
float yo[19] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

```

```

int i=0;
while (i < 5){
    yo[i] = min (y1, trapesium (xo[i],-50,0,-25,5));
    i=i+1;}

```

```

while (i<6){yo[i] = max (min(y1, trapesium(xo[i],-50,0,-25,5)), min
(y2, segitiga(xo[i],-20,25)));
    i=i+1;}

```

```

while (i<10){yo[i] = max (min (y2, segitiga(xo[i],-20,25)), min (y3,
segitiga (xo[i],0,40)));
    i=i+1;}

```

```

while (i<14){yo[i] = max (min (y3, segitiga (xo[i],0,40)), min (y4,
segitiga (xo[1],20,25)));
    i=i+1;}

```

```

while (i<15){yo[i] = max (min (y4, segitiga (xo[1],20,25)), min
(y5, trapesium(xo[i],25,5,50,0)));
    i=i+1;}

```

```

while (i<19){yo[i] = min (y5, trapesium(xo[i],25,5,50,0));
    i=i+1;}

```

```

float numerator = 0;
float denominator = 0;

```

```

for (int i=0;i<20;i++){
    denominator = denominator + yo[i];

```

```

        numerator = numerator + yo[i]*xo[i];
    }
    int z= (numerator / denominator);

    if (z>0){z= z*2.0;}
    else if (z<0){z = z*1.4;}

    ui->PWMver->setText(QString::number(z));

    pwmVer = z;

    return z;
    error0y = error0y;

}

```

Program pada arduino:

```

// include the library code:

#include <LiquidCrystal.h>

#include <SoftwareSerial.h>

static const int RXpin = 4, TXpin = 3;

static const uint32_t Baud = 9600;

SoftwareSerial ss (RXpin, TXpin);

long int dataH=0,dataV=0;

int penanda=0;

```



```

int datake=0;

char i;


#define pwma 6

#define in1 7

#define in2 8

#define in3 9

#define in4 10

#define pwmb 11

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, A0, 5, 4, 3, 2);


void setup() {

  Serial.begin(9600);

  ss.begin(Baud);

  pinMode(in1,OUTPUT);

  pinMode(in2,OUTPUT);

  pinMode(in3,OUTPUT);

  pinMode(in4,OUTPUT);

}

```

```

signed long nV = 100;

signed long nH = 100;

char RH[8];

char RV[4];

void loop()

{

while(Serial.available() == 0){ }

    Serial.readBytes(RH, 8);

ss.print ("H: ");

    ss.print(char(nH));

    ss.print (" V: ");

    ss.print(char(nV));

    ss.print("\n");

if (nH>255){nH=255;}

if (nH<-255){nH=-255;}

if(nH>0){

    analogWrite(pwmb,nH);

    digitalWrite(in3,1);// kanan

    digitalWrite(in4,0);

```

```

    }

else if(nH<0){

    nH=(nH*-1);

    analogWrite(pwmb,nH);

    digitalWrite(in3,0);// kiri

    digitalWrite(in4,1);

}

if (nV>255){nV=255;}

if (nV<-255){nV=-255;}

if(nV>0){

    analogWrite(pwma,nV);

    digitalWrite(in1,1); // maju

    digitalWrite(in2,0);}

else if(nV<0){

    nV=(nV*-1);

    analogWrite(pwma,nV);

    digitalWrite(in1,0); // mundur

    digitalWrite(in2,1);}

```

*# Halaman ini sengaja dikosongkan #*

## **BIODATA PENULIS**



Zakki Muhammad lahir di Jakarta pada 27 Desember 1994. Anak pertama dari 4 bersaudara. Penulis telah menempuh pendidikan formal yaitu di SD Pisangan 2 Ciputat, SMPIT Ruhul Jadid Tangerang dan SMAIT Al-Kahfi Bogor. Setelah menempuh pendidikan di SMAIT Al-Kahfi, penulis melanjutkan pendidikan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Di jurusan Teknik Elektro ini penulis mengambil Bidang Studi Elektronika. Penulis sempat aktif di pengurusan Unit Kegiatan Mahasiswa (UKM) Robotika, UKM bela diri Ju Jitsu dan bagian rohani dari Himpunan Mahasiswa Teknik Elektro.

E-mail:

zakkiwalker@gmail.com

*# Halaman ini sengaja dikosongkan #*